Restructuring Information in Technical Documentation when Migrating to XML-based Documentation System – A Case Study for Marioff Corporation Oy

Iida Koppanen
University of Tampere
Faculty of Communication Sciences
Master's Programme in English Language and Literature
Master's thesis
April 2018

Tampereen yliopisto
Viestintätieteiden tiedekunta
Englannin kielen ja kirjallisuuden maisteriopinnot

KOPPANEN, IIDA: Restructuring Information in Technical Documentation when Migrating to an XML-based Documentation System – A Case Study for Marioff Corporation Oy

Pro gradu -tutkielma, 66 sivua + 3 liitesivua
Huhtikuu 2018

Tämä tutkielma käsittelee tiedon uudelleenjärjestämistä teknisessä dokumentaatiossa siirryttäessä XML-pohjaiseen dokumentointiympäristöön. Teknisen viestinnän alalla XML-pohjaiset dokumentointijärjestelmät ovat yleistyneet 2000-luvun alusta lähtien ja monet yritykset ovat edelleen siirtymässä uuteen järjestelmään. Tämän tutkimuksen tarkoituksena on selvittää, miten tietoa kannattaa analysoida ja järjestää uudelleen siirryttäessä uuteen dokumentointijärjestelmään.

Teoreettisen viitekehyksen ensimmäinen osio koostuu informaatioarkkitehtuurin ja informaation suunnittelun välisen suhteen ja niiden eri piirteiden käsittelystä. Toinen osio käsittelee modulaarista dokumentaatiota ja siihen liittyviä periaatteita, kuten yksilähteistämistä ja DITA XML -kieltä teknisen viestinnän kontekstissa.

Tutkimus on Marioff Corporation Oy:lle tehty tapaustutkimus. Tutkimuksen metodina käytetään teoriaohjaavaa laadullista sisällönanalyysiä, jota ohjaa sisällön uudelleenkäytön analyysin malli. Uudelleenkäytön analyysin mallia käytetään, jotta analyysilla pystytään vastaamaan juuri tiedon uudelleenjärjestämistä koskevaan tutkimuskysymykseen. Tutkimuksen aineistona toimii Marioff Corporation Oy:n tekninen dokumentaatio, joka koostuu palonsammutusjärjestelmän osana toimivan pumppuyksikön käyttöohjeesta ja teknisistä datalehdistä. Tutkimuksen hypoteesi on, ettei Marioff Corporation Oy:n dokumenteista löydy mittavasti uudelleenkäytettävää sisältöä.

Tutkimuksen johtopäätös on, että uuteen dokumentointijärjestelmään siirrettävä materiaali on analysoitava huolellisesti yrityksen kontekstissa. Analyysin tulee tehdä henkilö, joka tuntee yrityksen tuotteet ja osaa täten arvioida, miten sisältö kannattaa jakaa moduuleiksi. Huolellisella analyysillä voidaan maksimoida sisällön uudelleenkäyttö. Tutkimuksessa esitetty hypoteesi ei toteudu vaan dokumenteista löytyy odotettua enemmän uudelleenkäytettävää sisältöä.

Tutkimuksessa selviää myös, että sisältöä pitää analysoida yrityksen sisällönhallintajärjestelmän asettamien rajoitusten puitteissa. Erityisesti sisällön uudelleenkäytön strategia pitää suunnitella sisällönhallintajärjestelmän mukaisesti. Tämä tutkimus tukee aiemmissa modulaarista dokumentaatiota käsittelevissä teoksissa esiintyvää ajatusta, että yksityiskohtaisia ohjeistuksia sisällön uudelleenjärjestämiseen ei voi antaa, koska sisällön jakaminen on tapauskohtaista eli riippuu jaettavasta sisällöstä, jota siirretään modulaariseen dokumentointiympäristöön. Tämän tutkimuksen analyysin pohjalta nostetaan kuitenkin esiin asioita, jotka kannattaa ottaa huomioon uuteen dokumentointiympäristöön siirryttäessä. Tutkimuksen jatkotutkimusaiheena esitetään palautteenkeruuprosessin muuttuminen XML-pohjaisessa dokumentointiympäristössä.

Avainsanat: modulaarisuus, tekninen dokumentaatio, tekninen viestintä, uudelleenkäyttö, informaation suunnittelu

Table of Contents

1 Introduction

Moving to XML-based writing and migrating documents into XML-based systems has been a trend in technical communication for the past decade (Hoffmann, 2012). The migration process is still ongoing in many organizations. In this process, technical documents, such as user manuals, are moved from traditional word processing programs, such as Microsoft Word, into a new XML-based documentation environment that has a different logic of producing and structuring content.

In XML-based documentation systems or *modular documentation systems*, documents are composed of smaller pieces called *modules*. A *module* ideally contains one topic that is understandable by itself but can also be used in a larger context (Bellamy, Carey & Schlotfeldt 2012, 8). Instead of being hierarchically built to have a clear beginning and an end, a document is built from small, related but individual pieces of information (Ament 2007, 6). These pieces of information can be used in multiple documents or in different contexts within the same document to make the content more unified and reduce the time spent on rewriting and managing almost identical content in various locations. This is called *reuse*, and it is one of the main benefits that modular documentation offers. Reuse is enabled by *single-sourcing*, which means storing information in one place, where it can be retrieved to be used in multiple contexts (ibid., 3).

In this study, I will discuss how the information in technical documents should be analyzed and divided into small reusable topics when moving technical documents from a traditional word processing program to an XML-based system, and what kind of problems might occur during the process. The main research question of the study is how the information in technical documents should be restructured when migrating to a modular documentation system. The migration process differs from one company to another. Thus, the purpose of this study is not to create a universal model of restructuring information for all situations, but to examine the challenges of documentation migration particularly in a company that produces hardware, and has a versatile set of documentation. The study is done as a case study for Marioff Corporation Oy (later Marioff), and it provides value for their process of migrating documentation into a new modular documentation system. I currently work at Marioff as a technical writer, and for that reason have previous knowledge of their documentation and valuable insight to carry out this study.

The purpose of the study is to form an outline for the new information structure while discussing the possible problems when moving documents into a modular documentation environment in the present organizational context. In this study, I will discuss the principles of *information architecture* which means the order and structure of information that makes the document easy to use (Ames & Corbin 2007, 12). *Information design* is a related term that refers to the presentation of information, such as layout (ibid.). Another central term is *DITA XML*, which is an XML-based language used in structured documentation. DITA XML is not implemented fully at Marioff in the early process of moving documentation into the XML environment. However, the principles provide a background for my analysis, since DITA will be implemented in the future and the purpose of this study is to provide a kind of optimized model of restructuring information.

Marioff has only a modest amount of technical documentation: less than 800 documents of which 80 percent are short documents. The documents are also in various formats, which is why an automatic migration process is not considered useful or cost-effective since the documents would still need to be vastly edited after the migration. Thus, the migration will be done manually, which is why it is important to make a well-thought information structure to support the benefits the new documentation system can offer. The suggested information structure outline provided in this study could also aid other organizations in similar situations.

Easier management and standardization of overlapping content, as well as achieving a more efficient translation process are some of the benefits companies wish to gain from XML and reusability. For the restructuring of the information, this means that the chunks of information included in modules must be optimized so that they are above all reusable, and do not follow chapter breaks unless the whole chapter can be reused. The problem that might occur if information design is done with reusability first is that the information becomes more and more general, and thus provides no information value for the user. One of the purposes of this study is to prevent such problems from occurring when planning reusability strategies.

The theoretical background of this study consists of works on information architecture and information design, as well as modular documentation, reusing content, and the basics of DITA XML language. For theory on information architecture and information design, I will use, for example a book by Albers and Mazur (2003). Ament's (2007) book will provide theoretical background for the discussion about modular documentation and reusing content. The principles and practicalities of DITA XML will be discussed through DITA XML guidelines by Bellamy, Carey, and Schlotfeldt (2012).

Similar migration processes have been done in various organizations, but the restructuring of information by content analysis has not been studied much from the practical point of view of showing how the content is divided and why. Koikkalainen (2002) made a related study for Nokia Mobile Phones focusing on the strengths and weaknesses of single-sourcing, and the process of migrating to a new documentation system. The study differs from my own by being made for a larger company with much more documentation and products that have similar features, which means that there are more possibilities for reusing content. The studies on organizations migrating to XML environments have also been studied from the point of view of the whole process, the benefits of modular documentation, and single-sourcing as a phenomenon (see for example, Broberg 2016).

My hypothesis for the study is that between different document types at Marioff, such as user manuals and technical data sheets, there is not much to be reused, because the information types included in these different document types are quite different. Currently, the user manuals include descriptive text and steps, whereas data sheets include mostly tables of technical information such as dimensions and voltages, which are not typically included in user manuals. At the moment, the location of different types of information is quite fixed, and overlapping content is not found between document types. However, information that is not currently in the user manuals but is included in data sheets could be reused in the manuals in the future, if users so require. An example of this might be service space reservation illustrations that are needed when installing a pump unit. These illustrations are currently in the user manuals. Restructuring documents might provide fruitful discussion in the organization about what information should be included in which documents. If my hypothesis proves true and there is not much to be reused, the value of the study is in the realization that the technical documents should be designed and produced in a different manner in the future so that the pieces of information are more reusable, especially if reuse is what the organization wishes to gain from the new system.

The data used in the study includes a user manual for operating and maintaining Marioff HI-FOG® Gas-driven Pump Unit (GPU) as part of the fire suppression systems, and four related technical data sheets. I chose this data because it is easily adaptable for the purposes of this study: it can be either expanded or delimited as needed by taking different aspects of the system into account or disregarding them. It also provides further analysis opportunities for the documentation of other Marioff pump units. The user manual has 49 pages, and the technical data sheets have approximately 3 pages per data sheet. The user manual mostly includes descriptive text, assembly images, and instructions. The technical data sheets include tables of technical information and figures with some descriptive text. The data will be acquired from Marioff.

My chosen research method is qualitative content analysis. Content analysis is suitable for this study because a close inspection of the content is crucial for the process of structuring information. Content analysis is about searching for differences and similarities in data sets (Tuomi & Sarajärvi 2002). It is exactly what I intend to do in this study. In my content analysis, I will divide the information into smaller pieces with the help of the theoretical background. I will utilize a five-step content reuse analysis by Bellamy et al. (2012). The five-step analysis consists of familiarizing oneself with the content being analyzed while taking into consideration the different purposes the texts may serve, identifying overlapping content, editing content to allow reuse, reorganizing content, and finally implementing the reuse strategies. Based on my analysis, I will suggest how the information should be restructured in the chosen documents and why, and how the restructuring could be utilized in the migration of other Marioff documentation.

This study will continue with an overview of information architecture and information design in Chapter 2. After that it will move on to the subjects of modular documentation, single-sourcing, and DITA XML in Chapter 3. Chapter 4 will introduce the case-study, materials and methods in detail. Chapter 5 will go through the analysis process and present the main results of the study. The conclusions of the study, with evaluation of the success of study, most important findings, and further study questions are presented in Chapter 6.

2 Information architecture and information design

The scope of tasks and skills assigned to technical communicators is constantly growing (Kimbal 2017, 345–6). Instead of focusing solely on writing and editing technical content from input given by subject matter experts, such as engineers and software developers, technical communicators are more and more involved in the overall planning of the structure and appearance of the content (ibid). For example, they can be involved in visualizing the content with images and layout design, as well as planning the information structure of the documents or even the whole information database. These tasks are usually listed under the fields of information architecture and information design.

In a world where the information load is constantly growing, creating quality content with suitable structure and presentation is more important than ever. The right information has to be found quickly and it has to be in an understandable form. According to Carliner (2003, 51), the cognitive load can be resolved for instance by communicating with images instead of text or stripping the content from unessential information. In successfully communicating information, the principles of both information architecture and information design become pivotal, as will be discovered in the following sections. The sections will discuss the principles of information architecture and information design, how they differ from each other, and what their relationship to technical communication is.

2.1 Information architecture

Information architecture essentially means the underlying structure of information, according to Rosenfeld and Morville (2000, 5). Just like the architecture of a house, the base of the whole building are the structural properties, such as a floor plan that defines how it is built from bottom to top. The floor plan is hopefully not designed in isolation though, but made to complement the function the house is supposed to serve. An apartment building most likely has a very different floor plan from an industrial building or a museum. Rosenfeld and Morville (ibid., 3) similarly use a house analogy to illustrate the complicated nature of information architecture. Rosenfeld and Morville (ibid.) state that all the different parts of the building have to work together. The house in information architecture can be, for example a database, library, or a single document. And as with designing a house, the function of the information should affect the structure. A novel with the same information structure as a newspaper with many unrelated topics on one page would be a confusing read.

Information architecture is a multi-disciplinary field that can intertwine with any other field that provides or handles information in one form or another. Rosenfeld and Morville (2000, 5) make a distinction between data, knowledge, and information: data provides answers to specific queries, knowledge is what people have, but information can be anything from images to articles that have more than one meaning. This means that information can be interpreted and used in different ways. Information architecture is applied, for example, in information technology, in user interface structure, and navigation on websites. Information architecture does not address the visual appearance of the information on a website for example, but the structures below the surface (ibid.). As technical communication deals with information in many shapes and forms, the principles of information architecture are also applied in structuring technical documents and documentation libraries. As will be discussed later, information architecture is also crucial in single-sourcing, which is utilized in modular technical communication.

Although information architecture is practiced in many fields of art and science, the core idea in all the fields is the structure of information. According to Crystal (2007, 16),

> [t]he information architect's role is to organize information appropriately by creating a hierarchical structure that is comprehensible to users, provide navigation structures that enable users to move through the information space, label categories and groups of content in sensible ways, and design search systems … that allow users to search for information effectively.

As is seen from the description above, the purpose of information architecture is to compose the larger picture and create logical structure, and aid the user to search and find the needed information. Crystal (ibid.) applies Rosenfeld and Morville's (2002, 5) ideas of information architecture's main components that create structure: organization of information, navigational structure, labeling content, and enabling effective search for information. Table 1 includes some of my own examples of what the four components could entail in a technical documentation framework:

Table 1 Examples of Rosenfeld and Morville's (2002, 5) four main components of information architecture

Component	**Examples**
Organization of information	– Content organized so that the first thing the user must do appears first in the document, the second appears second and so on – Division of content into smaller meaningful units that are linked to each other
Navigational structure	– Links and references – Table of contents – Page numbering
Labeling content	– Titles and headings – Metadata – Links between related documents and related pieces of information
Enabling effective search	– Index

The components are already intertwined, and for example metadata could be either under labeling or enabling efficient search, and table of contents could be listed under navigation or enabling efficient search. Although intertwined, the components must be actively made to work in unison for the information architecture to be successful. The user starts with searching for the information. If the content is labelled right and linked right, the search is easier and more efficient. If the navigational structure is logical, the information is found, and if the information is organized correctly, there is also a better chance for it to be understood.

Although Crystal's (2007, 16) description of information architecture is quite comprehensible with the application of the four main components, namely organization, navigation, labeling and search, part of the description in my opinion does not entirely fit the modern ways of organizing and consuming information. Crystal (ibid.) says that an information architect creates a hierarchical structure that is comprehensible to users. If we consider modular documentation, information is not necessarily hierarchical. The information architecture in modular documentation can be built to have a hierarchical structure, but even more the architecture establishes all the connections between pieces of information, labeling and navigational structure. Depending on the application of modular documentation, the user can assemble the parts into a comprehensive whole, into the hierarchy that serves their own goals. According to Sapienza (2004, 405), the hierarchy can be created by the user. Hierarchy is a suitable term, for example, in relation to the navigational structure of menu items on websites, because menus are often ordered hierarchically into levels. However, pieces of information do not always fit into a hierarchical structure, and when texts are more and more non-linear, there is not just one hierarchy.

Ames and Corbin (2007, 11–12) define information architecture neatly as the structure and relationships between pieces of information. This definition is in my opinion more sensitive to non-linear information than Crystal's definition. Information architecture helps the user make the connections between pieces of information, and perceive the larger picture, while providing them with the freedom to use the information in pieces. Good information architecture leads the user through the information to the goal they are trying to achieve, and if all related information is carefully linked, there are many paths to the same piece of information with freedom for the users who have different types of goals.

By establishing the relationships between pieces of information, the information architecture also lays the base for future content creation. Depending on the architecture it can either restrict or support content creation. To return to the house analogy, let us imagine if the bathroom of a house could only be accessed from the outside of the house. The first time you entered the house, you might think there is no bathroom, because from inside the house there is no visible evidence that the bathroom exists. First time you were there, someone would have to tell you that the bathroom indeed does exist, and explain how you can access it. Explaining how to get to the bathroom would also be more complicated than if the door was on the inside. To link this example to information architecture and content creation: the information for which the user is searching should always be accessible from the most logical place, and it should have an entrance point to the next piece of information without interruptions. If the information cannot be found in the logical place, the users may give up searching. When people constantly fail to find the information they need, but the content is already so layered and intertwined that the whole information structure cannot be changed (like tearing down the whole kitchen to make a door inside for the bathroom), writers might have to start adding further instructions on how to find the information, and work around the structural weaknesses. If the information structure is poorly designed from the start, it requires extra effort from both the user and the writer to make it work. If the foundation is laid with careful planning, it is easier to start building good quality on top of it.

To summarize, information architecture is the foundation of all information. Whether it is an online user guide, or printed assembling instructions for furniture, if information architecture is done with careful planning, the user can find the needed information, and understand its relation to other pieces of information. Creating content into the information architecture is also easier. In the next section, I will discuss a related discipline, information design, which deals more with content creation.

2.2 Information design

Albers (Albers & Mazur 2003, 2) starts his definition of information design by differentiating it from graphic design, web design, and information architecture. Schriver (ibid., ix) states that information designers have long focused on creating quality content, but much of the literature is still about visual design and typography. Visual design and typography are important parts of information design, and according to Albers (2003, 4–5), it is information design’s purpose to “bring together prose, graphics, and typography and make them work in unison to achieve the desired effect”. Thus, information design is much more than just visualization. Just like the interior of a residence building, if all the furniture is pretty, but the beds have no mattresses and the stoves do not heat, the house becomes quite useless for the purpose it was created. Albers (ibid., 1) defines high-quality information design as something that

> communicates information in a manner appropriate and pertinent to a reader's situational context. It must focus on the reader and ensure that he or she can clearly extract the information needed to accomplish the real-world goal which sent them searching for information.

As the above text suggests, the information should be understandable, and usable. It should also be relevant to the situation, and presented in a clear manner. This cannot be achieved by only designing the appearance of the information, but needs other aspects, such as certain language requirements that will be discussed later in this section.

Above I use the word *usable* to describe information. Usability is a large and complex field of study, and going into detail about usability research does not serve the purpose of this study. For this very reason, I will only discuss one of the layers, the usability of content, which in my opinion is a vital part of creating quality technical documentation. According to Redish (2000, 163), usable content is content that is not only pleasing to the eyes, but is easily found, understood and utilized by the user. Technical documents perhaps more than many other genres are used to find specific information to complete a specific task or solve a problem – preferably as fast as possible, according to Graves and Graves (2011, 33). Technical documents are not usually read from cover to cover, as is stated by Ganier (2004, 15). Instructions, in particular, are usually read simultaneously with trying to complete a task (ibid.). For this reason, according to Redish (2000, 163), it is especially important to make the information:

1. easy to find
2. simple and clear enough to be understood fast
3. relevant for the situation of the user.

Schriver (2003, x) says that "[i]nformation that can be retrieved in a few seconds and that looks "short and snappy" can be deemed useless if the content is ambiguous, abstract, leaves out critical detail, or is simply wrong”. With documentation for hardware, and especially for non-consumer products like Marioff’s, usable content is crucial. Inadequate or wrong information can delay the installation, or in the worst-case, damage the system, and cause later malfunctions. With products that are designed to protect people, it is vital that the installation is done correctly, and instructions are an important part of achieving that. Thus, all the three points mentioned above should be applied together to achieve optimal usability of content.

An important part of creating usable content is considering the actual user. Users have different levels of expertise and knowledge when they interact with products. Ideally the content is customized for different knowledge levels. According to Schriver (2003, xi), information for the novice users is often insufficient for users who have prior knowledge of the topic. Customizing can be achieved with XML-based documentation. This will be discussed more in Chapter 3. Customizing content to every level of users is not an easy task, first, because there are as many knowledge levels as there are users, and second, the knowledge levels may be hard to define without committing to an extensive user research to discover who the users are and how they use the product. It is not only the informational content and its detailing that has to be customized either, but also the language in which the information is conveyed. Knowing the user and their needs for the documentation, as well as creating content that suits those needs, should be one of technical communicator’s basic tasks and skills, since their work consists of acquiring information and then communicating that information to others in an efficient way that helps them complete tasks.

The information design process can be divided into different aspects as its final product is a complex combination of different forms of media, such as text and images, which takes into account all users. For example, Carliner (2003) divides the information design process into a three-part framework: physical, cognitive and affective design. Physical design deals with finding information, cognitive is about understanding information, and affective is about motivating users to perform (ibid., 46). The requirements of usable documents described earlier in this section, namely finding information, understanding it, and being able to utilize it, fall mostly under physical and cognitive design. Physical design shares similarities with information architecture, because it enables users to find information. I would make the distinction that information architecture works on the higher level of navigating through information, whereas information design uses for instance textual and visual cues to guide to the information. Further differences between information architecture and information design will be discussed in Section 2.3. Cognitive design, on the other hand, is also important in modular documentation and content reuse, because an individual piece of information has to be designed so well that it is usable in any context, according to Carliner (2003, 52). Modularity and reuse will be discussed in Chapter 3.

Affective design deals more with language and visuals that draw the user in and motivate them to complete tasks (Carliner 2003, 52–53). I would associate affective design more with documentation for consumer products where users would be expected to be more invested in the product and its functions. User studies done at Marioff suggest that the users of the documents are most interested in finding information as quickly as possible to complete the work. The installation work is also very segmented, and personnel changes constantly from one part of the installation to another. For this reason, users only search for the information relevant to their current task. Thus, affective design does not seem to have an immediate link to the documentation at Marioff, because it is enough to find and understand information as fast as possible to complete the tasks, and that can be achieved with physical and cognitive design.

There are some concrete guidelines that help in achieving a usable information design. One of them is basic technical writing and editing, which is under physical design in Carliner's (2003, 56) framework. I will not give a detailed description about everything that basic technical writing entails, but will introduce some issues that I think are most relevant from the point of view of information design. Basic technical writing entails guidelines such as using clear and concise language, writing in the active voice instead of the passive, and using titles and headings that describe the information that follows them. These basic guidelines are defined in the SFS standard for creating technical documents (SFS-EN 82079-1 2012). In addition to basic technical writing, layout and typographical decisions define the content's appearance. Guidelines regarding the layout include pages with enough white space that the reader is not overwhelmed by the content, and their eyes are steered to the right places on the page (Alred, Brusaw & Oliu 2009, 299). Using lists and tables can also help steer the gaze, and make complex information easier to process, because the relationship of information is presented visually (ibid. 519). Typographical guidelines, on the other hand, include for example line spacing, and font size (ibid., 296–298). The purpose of these guidelines is to help create usable content.

To summarize, information design is creating different types of information (text, images, and graphical elements) and combining them to form an understandable, functional whole. All the parts should function seamlessly both individually and together.

2.3 Differences between information architecture and information design

Information architecture and information design are two overlapping disciplines that share many similarities, but on a closer inspection have a different perspective, as was discovered in the previous sections. To return once again to the house analogy, information architecture defines the structural properties of the house, how the rooms are connected to one another, and what the overall function of the building is. It can even define the yard, and traffic signs pointing to the address. Information design, on the other hand, defines what the house looks like, what the interior looks like, how the items inside are organized, and most importantly whether the individual items in each room form a functional whole. Garrett (2001) differentiates information architecture and information design by stating the following:

> Information architecture is primarily about cognition … information design is primarily about perception … Information architecture belongs to the realm of the abstract, concerning itself more with structures in the mind than the structures on the page or screen. Information design … couldn't be more concrete, with considerations such as color and shape fundamental to the information designer's process.

The above distinction of information design being primarily about perception and information architecture in the realm of the abstract is in my opinion slightly oversimplified. As was discussed in the previous section, information design is often associated only with visual design, which leaves out important aspects, such as providing users with content that is usable and useful. Usable content cannot be achieved without considering the cognitive aspect of the user, which was established in Section 2.2. Even the layout design should consider the cognitive processing of information, for example white space, and minimizing cognitive load. The decisions made for the information design should be based on understanding how the human mind processes information, and what the best way to design the content at hand is. In Garrett's definition, information design actually taps into the realm of information architecture.

As we can see, the distinction is not always clear cut. In addition to having many things in common, information architecture and information design as terms are often used interchangeably. Some prefer information design, some information architecture even if they are talking about the same principles. Even Albers and Mazur's (2003) book *Content and Complexity: Information Design in Technical Communication*, with information design in the title, has articles about information architecture.

To further establish the relationship between the two principles, it is useful to look into the past of the terms. Richard Saul Wurman coined the term *information architect* in 1976, and with it established some of the main principles of information architecture (Knemeyer 2004). Wurman (quoted in Mazur 2003, 25) justifies his creation of the term *information architect* by stating the following:

> I selected the term information "architect" rather than information "designer" as the term "designer" continues to be interpreted by the public as an individual who is hired to come in after the fact to make some project "look better" – as opposed to a professional part of the initial team creatively solving a problem. I do not believe I can change this popular preconception. I believe the term information architect evokes rigor in the creation, research, choice as well as the presentation of information in an understandable yet artful form.

As is seen from the text above, Wurman's issue with the term *information design* is exactly what the discipline has suffered from in many studies since: association with visual design over creating quality content, as well as the lack of information planning as part of the whole process. Information architecture used to be synonymous with information design (Knemeyer 2004). The two disciplines, although sharing many things in common, have perhaps since moved further away from each other. Ames and Corbin (2007, 12) separate the two disciplines into different processes where principles of information design can be included in the process of information architecture, and offer the following description:

> Information architecture is an analysis and design process, whereas information design is a writing or development process. Information architecture involves the structure, organization, and retrievability of content, whereas information design involves the layout, design, and presentation of the content. Information architecture applies across chunks of content, whereas information design applies within chunks of content.

This distinction is in my opinion the most comprehensive so far. It includes the different sides of both information architecture and information design. Information architecture operates on the higher level of structure, organization and retrievability of information, whereas information design is about content creation whether it is written text, typology, or graphic items.

The two disciplines, separable but overlapping, should ideally be developed together (Ament 2007, 21). Similar to the structure of a house, and its function, content and its structure should be developed together. In technical communication, content usually has a specific predetermined function, such as a step list guiding through a task or troubleshooting section helping to find common problems. This function has to be taken into consideration in the design process. It would be useful for both information architects and information designers to consult each other to make the whole system work seamlessly. If the two are developed separately, there is a chance the information architecture defines the information design too much, as was illustrated with the bathroom example in Section 2.1. How well information architecture and information design can be developed together in practice is discussed in the conclusions.

3 Modular documentation

Modular text has existed for a long time in printed texts, such as newspapers and books that are composed of smaller pieces of information, articles and chapters (Bernhardt 1993, 159). Technical writers have used modular writing, as well, with word processing programs such as Adobe FrameMaker, where a document is composed of individual files that are organized into a book file. Modularity combined with single-sourcing and XML languages has become a trend in technical communication, because it has many characteristics that are beneficial to technical communication and its special requirements. Some of these characteristics were discussed already in relation to information architecture and information design, for example creating information that is quick to access and process. The benefits of modularity will be discussed in Section 3.2.

Modularity seems like a simple concept, but it actually entails many complex features. Sapienza (2004, 401) describes modularity in the following way:

> Modularization differs from traditional writing methods because it requires efforts that depart from linearity, hierarchy, and sequentiality. Modularity ensures usability by allowing individual text fragments to be queried and reused as needed by different media, purposes, and audiences. The fragments are not cut-and-pasted but rather re-sourced from a central repository.

Sapienza's description is a comprehensive summary of what modular documentation entails: non-linearity, single-sourcing, reuse, and usability of individual pieces of information in various contexts for different users. In 1993, Bernhardt took a look forward and imagined how the nature of text and our relationship to it would change when printed texts became history and on-screen texts took over. Purposeful division of content into smaller pieces started to appear with computer technology in the 1990s (ibid. 160). Technology has evolved massively from those times, but there are already interesting similarities with Bernhardt's ideas and the trend of modular structure of content in modern technical communication. At the beginning of the 1990s, the reason for modularizing text was the restricted space on a web page and computer screen (ibid.). In the beginning, content had to be split into smaller pieces often without considering the topical divisions (ibid.). This created a need to write content so that it was split into individual, understandable pieces. Bernhardt (ibid.) calls this *idea* *grouping*. The concept of idea grouping is exactly like topical writing in modular documentation. Topical writing, or topic-based writing, means writing pieces of information usually with a title and some content that covers only one point, according to Bellamy et al. (2012, 7). The point can be the procedure of turning on a coffee maker, for instance. In modular documentation, topics are created so that their content is general enough to be used on their own but they are also connectable to a larger context, as stated by Bellamy et al. (ibid., 8). The following sections will further discuss these issues related to modular documentation, namely single-sourcing and reuse, the principles of DITA XML language, as well as the main benefits of modular documentation in technical communication.

3.1 Single-sourcing and reuse

Essentially, *single-sourcing* means retrieving data from one single location to be used in various contexts (Ament 2007, 3). This means that the same piece of information does not have to exist anywhere else, because it can always be retrieved from that single location. Ament (ibid., 1) says that "[s]ingle sourcing is a documentation method that enables you to re-use the information you develop". Since the information is created only once, it has to be created in a way that allows it to be used in multiple outputs (ibid.). An output can be, for instance, a web page, a printed document, or a help guide in a mobile application. The same content can be published in any of these outputs.

In single-sourcing, the input (content) and output (format) are separated from each other in the information development process (Ament 2007, 3). Ament (ibid., 4) says that content is no longer created for a particular format, but to fit all formats. This means that the same content can be published in multiple outputs, and the form in which it is published does not affect the content itself. The appearance can be very different, but the information is the same. However, in my opinion, Ament's statement makes it seem like content is created in isolation from format, which sounds somewhat problematic. It sounds like writing without a context. In theory, there is a possibility to publish information in any form, but is the creation actually done completely in isolation from the format? Hackos (1999) says that in modular documentation "technical communicators must identify all potential uses of information, then prepare designs that accommodate all these different uses". This statement, on the other hand, compared to Ament's statement, makes it seem like information just exists, and the uses and their formats are designed later. In my opinion, the content has to have a function that is tied to the contexts it could appear in, before it can even be created. In the previous section, it was said that when creating modules, the content should be usable on its own but also as part of a larger context so that the topics do not become too isolated. Before content can be created as part of the larger context, questions such as what is needed, why it is needed, and who needs it, should be answered. With huge amounts of information in the library, and multiple writers in the team, knowing all contexts would become almost impossible, because all writers cannot know all products. In small writing teams, with a moderate amount of documentation, like at Marioff, knowing the contexts is much easier. There are ways to manage content in single-sourcing, such as metadata, which will be discussed later in this chapter. However, if we consider the separation of input and output only as the separation of content and the visual look, Ament's statement is much more acceptable. Still it makes me wonder if all content can really work in any context.

In line with Ament and Hackos, Eble (2003, 346) states that writers of modular text have to create "medium-neutral text", and Ford and Mott (2007, 336) say that content has to be universally usable, because it is used by diverse users in different formats. For the individual pieces of information, this means that the purpose and function have to be focused to be usable in any context. For the function to be focused, and the information to fit any format, the possible outputs, and most importantly users of information, have to be somehow predetermined to create usable content. If in the future Marioff's instructions are, for example, merged into software that guides users in the service work of the pump unit, in theory, the already existing modules can be utilized. What I wonder is though, if it is possible to make the modules so usable alone or to look so far in the future to make the information design suitable for all outputs, and not just think of the already existing designs. This subject will be discussed further in the analysis chapter. Sapienza (2004, 405) says that the writer must know their users or at least have a supposition of the users to create usable content by single-sourcing. Thus, a good knowledge of the different users is required if content is supposed to fit any format and be understandable to any user (see discussion in Section 2.2).

Although content should be usable in all contexts and user is one of the key aspects in the creation of usable content, in single-sourcing, all content does not have to be meant for all users — even if it is created to be universal enough to be understood by all users. Single-sourcing allows for customization by tagging and conditional processing. This means that content can be marked electronically to be targeted at certain audiences or outputs, according to Ament (2007, 18). When published, content can be filtered to include information only for expert users, for example. Tagging and conditional processing will be discussed in more detail in the next sections, in relation to DITA XML language, and metadata.

3.1.1 What is DITA XML language?

From the technical point of view, XML (extensible markup language) is what enables the user customization, reuse, and publishing in different outputs in single-sourcing. DITA (Darwin Information Typing Architecture), on the other hand, is an OASIS standard for XML. This means that the features included in DITA have been designed for a more specific purpose, as Linton and Brunski (2006, 7) state: "DITA was developed for designing, authoring, publishing, and managing content". In addition to this, according to Bellamy et al. (2012, 5), topic-based writing is specifically supported by DITA XML language. This means that when creating a module with DITA XML, the writer has to already choose the information type, or topic type. These topic types will be discussed later in this section. Perhaps because of these two reasons, namely that DITA was created to authoring content and it supports topical writing, DITA is one of the most commonly talked about markup languages in modular documentation in technical communication.

According to Bellamy et al. (2012, 229), "DITA is a semantic markup language, which means you must apply DITA elements to content based on what that content is rather than how that content should appear in output". For example, procedural information is put inside *steps*-element, because procedures are comprised of individual steps. Elements are smaller components of information inside the modules. The following is an example of an element that serves as the title of the module:

<title> Installation <title>

The content of the element that is to appear in the published document is written inside two tags. The individual tags start with < (smaller than) and close with > (larger than). The first tag marks the beginning of the element, and the second the end of the element.

Elements can be modified with attributes. With the attribute *blue*, the title can be modified to appear as blue instead of the standard color determined. Some attributes are only used to change the appearance of the output of the element, like text color or weight. Attributes are also used to change the function of the element. For example, a note can be modified to become a warning with the right attribute. The attribute *warning* will change the title and appropriate icon next to the content. Attributes are also used for marking content for different products and audiences. If an attribute is used this way to add further information about the element, it is considered metadata, according to Bellamy et al. (2012, 143). Metadata will be discussed further in the next section.

The appearance of the elements in the published output are controlled by a stylesheet. In the published output, the content inside *steps*-element might appear as a numbered step list with each number in different color if it is so configured into the stylesheet. Every document type or output has its own stylesheet that determines the appearance of the final product. Sapienza (2004, 400) says that stylesheets in XML are powerful because they can transform documents to any format or even to different languages. At Marioff, the stylesheets were created based on existing content and existing document templates with some modifications. The stylesheets were created with consultants at the same time as discussing the requirements for the new system. Thus, in this case, the format has been predesigned and content will be divided to more or less fit the existing formats. This does not mean that content is divided to fit only the existing formats, but to be as universally usable as possible for any new formats in the future.

Based on the above description, DITA seems different from traditional, non-XML-based writing, but it also seems to be a fairly simple way to create content. However, Bellamy et al. (2012, 17) say that

> DITA provides a flexible, yet rigid, framework that helps you create effective technical information. DITA is flexible in that it helps you reuse, reorganize, and create content quickly. It's rigid in that you must be disciplined enough to adhere to the principles of good topic-based writing to take advantage of the benefits of DITA.

What the rigidness of DITA means is that content has to be, first, created with the right elements, and second, created with the right information types. These information types are different kinds of modules: concept module, task module, and reference module. According to Bellamy et al. (ibid. 11), the three module types have their own functions: Concept is for descriptive content, task is for procedures, and reference is meant for any additional information users might need to complete a task. Johnson (2012) uses a figure to illustrate what kind of information the three module types usually include (see Figure 1).

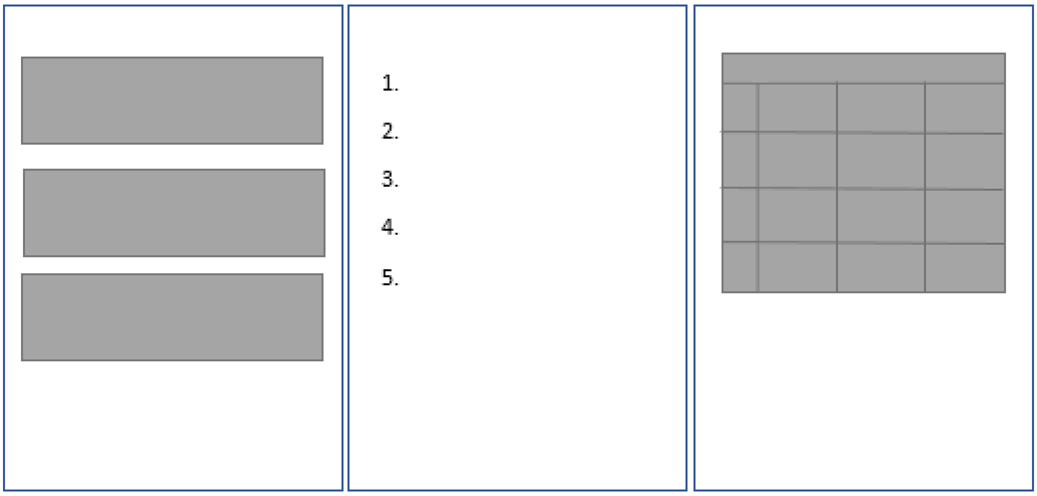


Figure 1 Concept, task, and reference (reproduced from Johnson 2012)

The first page illustrates a concept module that usually includes text bodies. The second page is a task module, which includes step procedures. The third page is a reference module, which can for example include a table of spare part information. The three modules in DITA include different elements, and only certain elements are allowed to be used in certain modules. There are also restrictions on the order in which those elements can be inserted. These restrictions guide the authoring so that the features designed into DITA are implemented correctly, and most of the benefits are gained through this correct implementation. Full DITA with all the included elements will not be implemented yet at Marioff, because of the rigidness of DITA. However, content will be divided into modules according to the three module types.

3.1.2 What is metadata?

When all the information is divided into small pieces, there is a lot more content to manage. For this reason, there has to be a way to find all related information easily. Metadata is used to label and categorize information to make it easier for the writers and users to find, as well as to allow targeting content for different audiences, according to Bellamy et al. (2012, 143). Metadata is in fact information about information (ibid.). Designing and implementing metadata use is information architecture, because it is part of the underlying structure of information that enables effective search, for instance (see Section 2.1). As was said about information architecture, which is the foundation of content, metadata usage also needs to be well-thought from the start to provide the full benefits. Metadata is not as fixed as the whole information architecture since it is only one part of it. However, careful planning is needed to avoid any large changes in the future. According to Bellamy et al. (2012, 144), the correct use of metadata makes it easy for:

1. users to find information, because keywords and index entries are found by the search engine
2. writers to manage information, because content is well organized and classified in the content management system (CMS)
3. writers to include and exclude content based on, for example audience, product or version.

For example, when beginning to gather modules into a document, the writer must first find all related content from which to choose the right pieces. The writer can search for all modules related to a certain product, for example, and if the metadata has been inserted correctly, all related content is found, and can be further filtered to narrow content down to the needed pieces of information. Users, on the other hand, find content they were looking for, and can also find content that has been targeted to their knowledge level. If the metadata has to be changed, especially in cases where users have already learned certain searching habits based on the metadata, the change could temporarily decrease usability.

At Marioff, there is a restricted set of metadata elements and attributes, because of the partial implementation of DITA. The metadata is designed to contain information about creation dates, version, document ownership, and so on. According to Bellamy et al. (2012, 150–3), these are called either topic metadata or map metadata, depending on where the metadata is added, the topic module or the map file. A map file is where all the other modules are gathered to create a document (ibid., 91). In addition to those metadata attributes mentioned above, there are certain attributes relating to audiences and products that can be used to publish customized documents.

Customizing documents by filtering and targeting content can be achieved with metadata designed for conditional processing. In conditional processing, the content is marked with metadata to include information about a certain version of a product, or to be targeted to a specific audience. When the document is published, this information can be filtered in or out of the published document (Ament 2007, 183). This allows for publishing different versions of the same document from the same map file. Thus, content does not have be maintained in two separate locations. This is useful only if most of the information in the documents is shared between the versions or audiences. If documents include more version specific content than mutual, the mutual content can be reused in the different versions by using content references (conref), according to Bellamy et al. (2012, 178). Content referencing is done by giving an element a unique ID (ibid.). With this ID as a reference, the element can be included in any module, in any context (ibid.). In other words, content referencing is one method to reuse content. Because of the restricted set of metadata at Marioff, it will not be discussed further in the analysis chapter. However, conditional processing and content referencing will be discussed because they enable some of the main benefits of single-sourcing, namely customization and reuse.

3.2 Benefits of modularity in technical communication

Modular documentation has many benefits in technical documentation. These benefits can be divided into benefits for the writers and benefits for the users. Bellamy et al. (2012, 5) summarize the benefits of modularity in the following way:

> For users of your information, well-written, topic-based information improves retrievability, navigation, and usability. For writing teams, effective topic-based writing provides more opportunities for reuse, quick reorganization of information, easier file management, and more flexible linking.

In the summary above, the improvements assigned for users are very similar to what was concluded to make a successful information design in Section 2.2: information is easy to find, understandable and relevant. For writers, Bellamy et al. (ibid.) list reuse, reorganization of information, file management, and flexible linking as the main benefits. These, on the other hand, are all more or less related to information architecture, specifically the structuring that occurs behind the content. The benefits a company wishes to gain from modularity might even affect the restructuring of information in the migration process. A company might want to prioritize some benefits, such as reusability or coherence, and analyze and restructure their content in accordance to the set priorities.

Below, in Table 2, I have listed some main benefits, and some related, more concrete benefits. The benefits have been gathered from Bellamy et al. (2012) and Ament (2007). The concrete benefits have been categorized into benefits for writers and benefits for users.

Table 2 Benefits of modular documentation for writers and users (based on Bellamy et al. (2012) and Ament (2007))

Benefit	**Writers**	**Users**
Effectiveness	Focus on one topic at a time Fast changes Reuse Quick reorganization of content Metadata Flexible linking of content Information management Layout Multiple writers Review process Localization	Clear, well-written content (usability) Correct information Faster changes Quick search Easy access to needed information Finding related topics easily Translations available
Coherence	Reuse Language and style (team synergy) Terminology Layout Localization	Language Terminology Layout Usability
Customization	Quick reorganization of content Targeting content for user groups	Customer-specific documents Relevant information Targeted content

First of the main benefits is effectiveness, the second is coherence, and the third is customization. In the following sections, I will discuss the benefits in more detail. As is seen from Table 2, some of the related benefits for writers and users are overlapping. They will be discussed in each subsection from the point of view of the main benefit.

3.2.1 Effectiveness

Effectiveness is the first main benefit under which majority of the other benefits for writer and users can be placed. First of the related benefits is the writer's ability to focus on one topic and one subject at a time, which in return leads to users receiving clear, well-written information. Previously in this chapter it was established that when focusing to write one topic at a time, writers are able to create content that is more usable. Usability, in return, requires well-written content, as was discussed in Section 2.2.

The popularity of modular documentation in technical communication can perhaps be explained partly by the need to be concise and clear, as well as for the information to be easy to process. Writing individual topics as self-contained units increases effectiveness for users, because according to Bellamy et al. (2012, 8),

> most users need only a small amount of information at a time, you should create information that can answer specific question discretely without requiring extensive reading across large amounts of interconnected content.

Writers are able to write better quality content, because they can focus on answering one question at a time, and users receive more usable, clear content that undertakes one task at a time. This concise writing aids users who typically have their specific goals that they need to fulfil with the information they acquire from the document, as is stated by Ament (2007, 14). Modular documentation supports these needs by usable topics that are written for any context. According to Ament (2007, 3), the writer can focus to make that topic as clear and usable as possible. According to Bellamy et al. (2012: 183–4), information is more likely to be accurate as well, when writers are able to focus on one topic at a time. Correct information is an obvious benefit for the users. For the company and the writers, it is also beneficial that incorrect information can be corrected fast and it will be corrected with one change into all documents where it occurs. These fast changes are achieved with reuse, according to Bellamy et al. (ibid.).

Reuse makes writing effective for several other reasons besides fast error correction. New documents can be assembled from the existing content when writers do not have to start from nothing.

In addition to this, since many companies have products that have similarities with other products made by the company, according to Bellamy et al. (2012, 183), it is more efficient to reuse those information units rather than rewrite them over and over. For example, smart phones made by the same company share a majority of the main features, so writing instructions for all the products separately would be redundant and inefficient, especially if the content has been written well once already. At Marioff, these different products where reuse could become very useful are the pump units or valves that share a lot of similar features, not to mention all the general information that is shared between products, such as safety information.

As was discovered in Section 2.2, users of technical documents want to find the information fast when they are completing tasks. In modular documentation, metadata usage and linking are used for this purpose. They enable quick search, easy access to information, and finding related content for the users. Metadata aids writers and users in different ways as was discussed previously in this chapter: writers use metadata to find related information to build into documents, and label content; for users, metadata helps to find correct information and find related information.

Modular documentation with reuse, metadata and linking also enables more efficient information management for the writers, because all content is in the same location where it can be easily retrieved. Metadata and a single location for all content aids in managing the growing information load. For example, at Marioff, files are no longer on personal drives, but on a common server marked with related metadata. This helps with document version management as well. Individual modules, in other words individual pieces of information all have their own version, which can be tracked. This allows for monitoring that users have the most recent version of the documents in use, which also means that the information the users have is still valid.

Effectiveness is also achieved because technical writers are able to focus on content instead of focusing on tweaking the layout (Ament 2007, 3). Layout comes from a separately designed stylesheet as was discussed in relation to DITA XML. Information design from the point of view of graphic design and typology is done outside the authoring process, since input and output are separated from each other. However, the information design done at the sentence and the terminology level is still done when creating the content, as well as combining different media, such as text and images, into a usable whole. According to some, for example Kimbal (2017, 338), separating content and format decreases the creativity of technical writers. However, according to Evia, Tech and Priestley (2016, 24), technical writers are still using creativity by explaining complex ideas to the users in an understandable way. Thus, moving to modular documentation does not eliminate all creativity from technical writing. But of course, for some technical writers who have enjoyed visualization as part of the job, modular documentation might seem less creative. At Marioff, the writers would rather leave visualization outside the writing process, so losing creative power is not an issue in this case. One of the benefits Marioff writers actually hope to gain from the new documentation system is that they can focus on content and not the layout.

Another benefit of modular documentation for writers is the possibility for multiple writers to work on the same document at the same time (Bellamy et al. 2012, 9). Ament (2007, 8) also lists this increasing team synergy as an important reason to start writing modular documentation. Instead of the whole document being unavailable for other writers during the modification, writers can work on different parts of the same document at the same time, as is said by Bellamy et al. (2012, 9). In other words, separate modules are not locked for different users. In addition to this, working on different parts of the same document could mean that different writers can have different responsibilities: one can be responsible for safety information, and the other maintenance. Working on different parts of the document also allows for a more efficient review process. People who review documents do not have to read a full document, but they can review one piece at a time, as is suggested by Bellamy et al. (ibid., 10).

Modularization not only makes the work easier for writers in the same team, but also makes localization of content more efficient (Bellamy et al. 2012, 183). First, because XML enables automatic translation of recurring elements with *lang*-attribute (Haynes 2018, 59). For example, texts such as “Table of contents” and “Warning” are recurring and unchanging information that can be translated with XML instead of being translated separately by a translation agency. Second, it is possible to send only the updated parts of a document for translation, instead of the whole document. In addition to being efficient, this also reduces translation costs, and makes the translations available to users faster. Translation management is another one of the benefits Marioff hopes to gain with the XML based system.

3.2.2 Coherence

Coherence is the second main benefit of modular documentation. Although coherence includes fewer smaller related benefits, it is fact, in my opinion, one of the most important benefits from the point of view of usability, because it relates to how information is processed, namely the cognitive aspect of information design. The cognitive aspect in the three-part framework of information design by Carliner (2003, 46) deals with creating information that is understood by users (see Section 2.2). In other words, it is the psychological aspect of how users process information. Coherence is an important part of that, because according to Eysenck and Keane (2000, 92), human brain learns to recognize patterns. In the context of technical communication, this means that users are able to find and process information faster when the content is coherent and thus its patterns are familiarized quickly. These patterns can be for example headings that are always written in imperative mood.

Coherence affects the writer and user in similar ways. Writers are able to create more uniform content, when they can reuse modules. Users, on the other hand, in effect receive more uniform content. This affects language and terminology, but also layout, which is controlled by a stylesheet. The same document types always have the same form because of the stylesheets. Coherence adds to the usability of the content for the user, because terminology, language and visual cues used in the document become familiar quickly and are thus more easily understood. Requirement for consistent terminology is expressed in the SFS standard (SFS-EN 82079-1 2012, 57) by stating that "[t]erminology shall be used consistently according to an editorial style guide thus enhancing comprehensibility”. For example, at Marioff, when users use technical data sheets, it makes finding information easier, if the documents are all structured the same way, and specific information can always be found from the same place, and by the same term. If users search for voltage information, it is much easier to find the information if it is always found on the first page, in a table, marked as *operating voltage*. Rather than the information changing place from one document to another, and being labeled *operating voltage* in some documents, and *maximum voltage* in some. These kinds of issues with inconsistent terminology will be addressed in the analysis section, when marking content for editing. There are benefits in coherence for localization as well. Terminology stays coherent also in translations, because the translation memory can match terms more accurately when there is no variance, according to Bellamy et al. (2012, 232).

However, for content to stay coherent, there have to be style guides so that writers follow common rules when creating content (Ament 2007, x). A style guide includes standard writing rules for all writers in the team to follow, and should not be confused with stylesheets that control the format of the documents. These writing guidelines can be capitalization of titles and use of punctuation, for example. Style guides are another way to increase team synergy. Ament (ibid.) claims that writing guidelines are even more important in modular documentation than in linear documentation, because content has to be usable in any context. This statement is very accurate, in my opinion, because the writing style affects reuse possibilities. When moving to modular documentation, the style guide should also include instructions for metadata use and content referencing, as well as file management. Even when modular documentation promises many great benefits, it is important to move content to the new system in the correct way, and keep creating content with common guidelines. In the process of moving documents to the XML environment, the legacy documents, for example their structure and layout, still easily haunt the new documentation. The mistake of copying and pasting without giving enough thought to the new ways of documenting is easy to make, especially with project time limits and other pressures to do the migration as quickly as possible. At Marioff, the style guide development is in progress, so some rules could be created based on this study, for example relating to the reuse strategy. I will address some of these style issues when analyzing the material.

3.2.3 Customization

The third and final main benefit is customization. There are different ways to customize content in the modular documentation environment. I will discuss two, namely reorganizing content, and conditional processing. Customization is one way to ensure that the third criteria of usable content, namely relevance of information to the user, is met (see SectionInformation design2.2).

Modularization enables easier reorganization of content, as is stated by Bellamy et al. (2012, 5). This means building separate documents for different user groups into separate maps, and customizing the structure and contents based on user needs. This is perhaps not the ideal way to customize, because two separate maps need to be maintained. However, it works in some instances. For example, all GPU installation instructions are project-specific. Project managers at Marioff create the project specific manuals using old project manuals as basis. With the new modular documentation system, these project specific documents could be assembled from existing modules, and completed with any new information that is required for that specific product. Since project-specific content is usually content that has restricted access, it cannot be combined with generic documents to be filtered out with conditional processing, which is why separate map files are a good option in this case. Customers receive customized content, restricted information stays safe, and content is coherent also across product-specific documentation.

Metadata usage and conditional processing are an efficient way to customize and target by marking content to include information about a certain product, project, or user group, and filtering that content into or out of the document (see Section 3.1.2.). Customizing can be done for different users, such as novice and expert users, but also for different outputs. For example, content can be different for the printed manual, and the online manual, because of the different ways to use these formats. However, customizing requires user testing, first, because the different user groups need to be identified to create appropriate metadata and filters, and second, to acquire knowledge of the ways users use the content. According to Dyke and Mott (2003, 334), in single-sourcing, including users in the information design and content creation early on in the process is very important, because the content can then be reshaped accordingly. The study results can also be utilized in designing new outputs. Customization can be done for the different document types, but also to layouts with the different stylesheets. For instance, this allows for changing a color scheme to be more appropriate for different cultures. This kind of customization could be considered also at Marioff since their customers are around the world, from very different cultures.

4 Material and method

This chapter introduces the case-study in more detail, presents the material of this study, and presents the method used to analyze the material.

4.1 Case-study – Marioff Corporation Oy

Marioff is a company specializing in fire-suppression systems that use high-pressure technology to create water mist. Their products are used to protect human life and property. This gives the technical documentation a special importance, because correct installation of the system is vital, and both experienced personnel and beginners should be able to complete installation with the help of the technical documents.

The documents have a special role also for not being written for the end user of the product, but for the installers, maintenance personnel, and project managers who supervise the installation process. The spectrum of users may range from very experienced users, who only come to the document to check some technical specifications, to someone who has completed training but does not have the routine of installing a system like HI-FOG®. For these reasons, Marioff documentation offers interesting research material for studying what kind of content should be created, or in the context of this study, how the content should be modified and restructured to be made usable in the new documentation system, as well as how the documents could be targeted at different types of users and uses in the future.

The technical documents at Marioff have different security levels for information: some are restricted for internal use, and some are available for everyone. There are also two business units, land and marine, that have their own requirements and needs for products and their documentation. These factors provide interesting development possibilities for technical documentation from the point of view of reuse and document customization.

The documentation library is stored in Windchill, which is a content management system (CMS) provided by PTC. The CMS was not developed for documentation management, but product lifecycle management. Product lifecycle management means that the different stages of the product, from design to production are reported and reviewed in the system. From the point of view of information architecture, it is not an ideal situation that documents are treated the same way as products. Very different kinds of content are stored, linked and go through processes the same way. This causes problems, for example for the retrievability of information, because search results consist of both products and documentation, which can be confusing especially for new users. The CMS also sets some restrictions to the new XML-based documentation system, because metadata has to be designed and configured to fit the already existing metadata in Windchill, which does not always support documentation. However, there are also some benefits in having documents and products in the same CMS: documents can be linked to products and their 3D models. This enables finding all related documentation under the products themselves, and linking images straight to the 3D model of which they were made. This enables tracking changes in products that need to be updated into the documents. These restrictions and benefits should be taken into consideration when restructuring information, and considering for example the reuse strategy. This will be discussed further in the analysis.

The documentation software that is currently used at Marioff is Adobe FrameMaker. As was discussed earlier, FrameMaker allows creating structured content. With longer documents at Marioff, content has been structured to chapter level, but not to paragraph or sentence level which will be done with the modular documentation system. This means that the content has to be divided without previous divisions as a guideline. The XML-based system that will be used at Marioff is called PTC Arbortext. It was developed by the same company as Marioff's current CSM system, Windchill. Since therse two systems are linked, it might make the information architecture and information design processed work better together.

The main benefits Marioff wishes to gain from the new system are focus on content over layout, reusing content as much as possible, making translation process more efficient and reducing translation costs, as well as preparing for any future endeavors of modernizing documentation. These benefits will be taken into consideration in the analysis, in addition to which I will modify content to increase the other benefits discussed in Chapter 3.

4.2 Material

My material consists of two technical document types from Marioff: Operating and maintenance manual, and technical data sheets. The operating and maintenance manual as a document type at Marioff usually includes a description of the product design, and procedures on how to operate, service and test the product. It includes mostly descriptive text, procedures, and images. Technical data sheets are used for communicating technical specifications such as voltages and dimensions. They are also sometimes used as sales material at Marioff. Data sheets mostly include tables of technical information, figures, and some descriptive text (see an example of Marioff data sheet in Appendix). In this study, I will analyze one operating and maintenance manual, "Gas-driven Pump unit – Operating and Maintenance Manual", and the following four technical data sheets:

- GPU with wall mounted cylinders
- Gas Driven Pump – Dual GPU
- Gas driven Pump Unit GPU – GPU-6
- Water tank positioning for GPU

The operating and maintenance manual is 49 pages, and the technical data sheets are 2–6 pages per data sheet. I chose to analyze two different document types to get a more comprehensive idea of the reuse opportunities across different documents.

The chosen documents are all related to HI-FOG® Gas-driven pump unit (GPU), which is one of Marioff's oldest products. It is a pump unit that provides water for the fire-suppression system. It utilizes pressurized air or nitrogen to reduce the droplet size of the water coming out of the sprinklers, thus providing a more efficient system that causes less damage to the protected area if the fire protection system is activated. I chose GPU documentation for two reasons: first, GPU is an old product, so there is a lot of legacy documentation to analyze and it is interesting to discover how uniform the documents have been kept; second, Marioff has five different kinds of pump units, so there is a chance to broaden this study to the other units. The pump unit documentation also has different levels of information: the individual parts that form the pump unit, the pump unit as a whole, and also the system level of all the other components that are needed in addition to the pump unit. For this study, this means that there are various ways to structure the documents: from the individual parts to the system level. The information has to be divided so that the documentation for different options of the pump unit, for example different motor options, can be combined from the pieces of information. This complexity of GPU as a product has its challenges but also opportunities for analyzing the documents and developing a new information structure for the documentation.

4.3 Method

My chosen research method is qualitative content analysis. In qualitative content analysis, the chosen material is studied to discover patterns that relate to the research question (Tuomi & Sarajärvi 2002, 112–13). To analyze the content, a unit of analysis is defined (ibid.). In my study, I will identify topics and analyze their relationship to each other. The unit of analysis in my study is thus an individual topic that can vary from one sentence to a paragraph or even a paragraph and a table depending on what is needed to form the topic.

In qualitative content analysis, the results are reported verbally, instead of providing numbers or percentages as results. Quantifying the results, when the data set is small, as it often is in qualitative research, does not bring further value to the study, according to Tuomi and Sarajärvi (2002, 119). However, I will quantify some of my results, namely the number of modules produced, and how many of those modules are reusable across the material of this study. Providing numbers of reusable content in this study does not give any indication of the complete reusability of Marioff documentation, because the material consists of documentation for only one product. It does, however, give an indication of the reuse potential.

There are three approaches to qualitative content analysis, according to Eskola (2015, 188): *material-driven* (inductive), *theory-guided,* and *theory-driven* (deductive) content analysis. In material-driven content analysis, the analyzable material is utilized to create new theories and terms to describe phenomena (Tuomi & Sarajärvi 2002, 97). Theory-driven content analysis, on the other hand, means analyzing the material using a pre-existing theory or model (ibid., 99–100). In my study, I will apply the theory-guided approach, which is between the other two approaches. Theory-guided content analysis means using a theory only as a guidance for the analysis, not trying to prove or disprove the theory, but only acknowledging the previous studies and using pre-existing terminology to explain the results (ibid., 97–8). When moving to an XML-based documentation system, there is not just one right way to do it, but the content itself that is being moved to the system affects the process. Bellamy et al. (2012, 196) state that there is no tool to identify all reusable content and move it ready to the system. However, there are certain technical documentation standards that need to be followed, such as the SFS standard, and there are guidelines for migrating to XML that are beneficial, such as guidelines provided by Bellamy et al. (2012). For these reasons, neither material-driven, which would mean restructuring without any guidelines, following only the characteristics of the material, nor theory-driven content analysis, which would mean strictly trying to fit the content to a certain model, would suit this study.

The stages of theory-guided content analysis according to Eskola (2015, 194–8) are:

1. Gathering material and choosing what to search for
2. Rearranging material thematically
3. Making conclusions with the help of the theoretical framework
4. Reporting most important or most interesting findings with connections to theory

Stage one has been already been concluded: material is gathered, and from the material, I will search for similarities for reuse, for example. In stage two, I will analyze the material, but instead of rearranging the material thematically, I will rearrange it into topics. For this stage, I will use another model to guide the analysis. Stage three will include comparing my results to the points presented in the two theory sections. In stage four, I will report my findings. The stages Eskola lists are quite general, but since content analysis can be used to analyze different types of material, they form a frame for the content analysis.

A model I will use as a guideline for the second stage of Eskola's content analysis, is the reuse analysis by Bellamy et al. (2012; see Table 3). This model will help to specify the content analysis to the current research question, which is restructuring information when moving to an XML-based documentation system. The book by Bellamy et al. (ibid.), which includes the model, was made for organizations that are moving to XML-based systems, and starting to use DITA XML language for the first time. The book was published by IBM, who developed the DITA XML language (Evia et al. 2016, 24). As was stated previously, restructuring information into an XML-based system is dictated by the content itself, and giving detailed instructions how to do it would be impossible. The reuse analysis step list offers a starting point for analyzing content to reuse, although it is quite general. It consists of steps from going through the material with a toothcomb to implementing DITA XML to the content, and finally publishing it.

Table 3 DITA Best Practices reuse analysis (reproduced from Bellamy et al. 2012, 197)

Step 1: Analyze Your Content Analyze the information set or library to understand the content. When you're selecting content to reuse, consider scenarios and business goals that have common task flows.
Step 2: Identify Duplicate and Near Duplicate Content Identify content that overlaps: • Is the content exactly the same? • Does the content have the same meaning but is written slightly different? • Is the content nearly the same except that a few pieces of content are specific to a particular product, object, or technology?

Step 3: Address the Duplication
Consider the following strategies for creating reusable topics: • For content that is the same except written differently, rewrite the content to make it identical. • Rewrite content to make it more generic. • Use conditional processing in topics that have similar content so that you can reuse the topic and exclude content that is specific to a particular audience, product, or version.
Step 4: Reorganize and Rewrite for Reuse
To rewrite for reuse: • Create reusable components. If necessary, separate content into more reusable topics and submaps. • Consolidate duplicate content by combining common in a single file but applying conditional processing values to different content that is meant for specific information set. • Use one DITA map for a specific component, area, or main user task, such as installation, security, hardware configuration, or application development.
Step 5: Implement the Reuse Strategy
Insert the reusable elements, topics, and DITA maps into your information sets. Use the copy-to and conref features and ensure that you're creating effective topic-based content.

Parts of the step list are not applicable to my analysis. For example, Step 5: Implement the Reuse Strategy goes beyond the scope of this study, which is also the case to some extent with Steps 3 and 4, because they already include actions to move content into the modular system. My study will only go as far as to provide a structure of the form in which the content will be taken into the system.

Since the step list by Bellamy et al. (2012, 197) was written to guide the reuse strategy, it does not include all steps required to completely restructure information, which is the goal of this study. For this reason, I will add, rearrange and reinterpret steps to make the step list more relevant and useful for the purpose of this study. I will start the analysis by first familiarizing myself with the material and mirroring the content to the different contexts where it could be used, as is suggested by Step 1. In addition to this, I will already begin to divide the content into individual topics, because it guides actions performed in the next steps. In Bellamy et al.'s step list, dividing content into topics does not come until Step 4, but since restructuring is the goal of this study, I will move it earlier in the analysis. After that, in Step 2, I will identify duplicate content and any parts of the text that need to be edited to combine into reusable topics, or to make text more generic. I will also create a reuse strategy for duplicate content. By moving the identification of individual topics from Step 4 to Step 1, and editing duplicate content from Step 3 to Step 2, Steps 3, 4 and 5 can be disregarded, because they do not apply to the scope of this study. Below is a modified table of the analysis process with rearranged order and concrete actions to be taken in every step.

Table 4 Modified reuse analysis

Steps	**Actions**
Step 1: Analyze your content	Go through material Mark potential topic divisions
Step 2: Identify duplicate content and content that needs editing	Mark duplicate content Mark content that should be edited Create a reuse strategy for duplicate content

In the next chapter, I will go through my analysis following the structure presented in Table 4. After completing the analysis, I will continue to present my results and discuss the implications of the study.

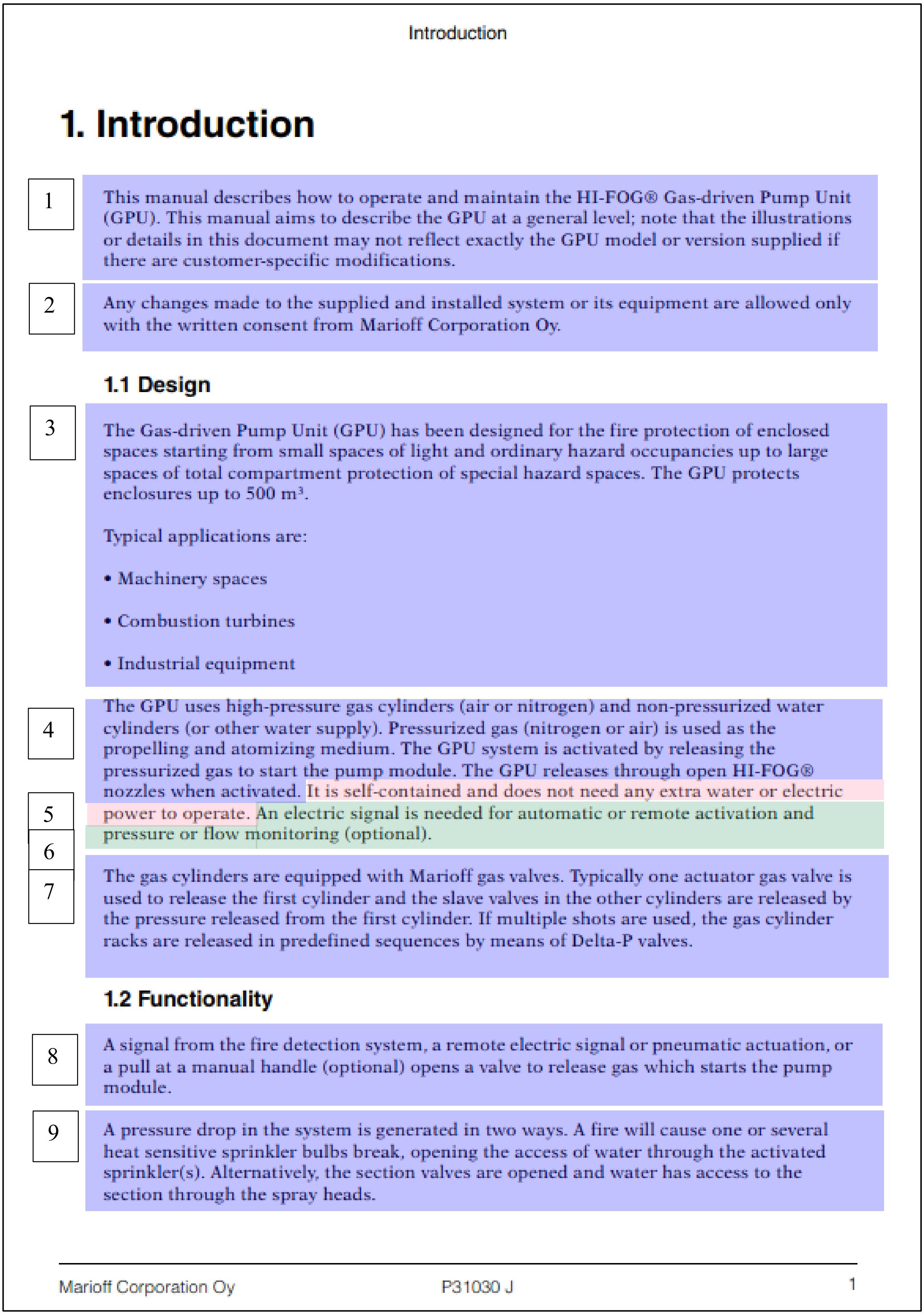
5 Analysis

In this chapter, I will give a detailed description of my analysis, introduce my findings, and illustrate the suggested method for analyzing content and diving it into reusable topics when moving to an XML-based documentation environment. The speculated reuse possibilities outside the material analyzed in this study relies on my prior knowledge of Marioff documentation. In this chapter, I will also discuss the possible problems and challenges that occurred during the analysis of the material.

First, I will go through the analysis process step by step, starting with identifying topical divisions, continuing with identifying duplicate content and content that needs editing, and ending with the solutions I have come up with the restructuring of the information. In my analysis, I have ignored sections that come from the stylesheet in the Marioff configuration, including copyright information that is automatically printed on the first page.

5.1 Identifying topical divisions (Step 1)

Early in the analysis, it became clear that dividing content into topics cannot be performed without already considering the reuse possibilities. If maximum reusability of content is the goal, reuse will affect the divisions. In Figure 2, I have illustrated a topical division of one page from the GPU Operating and maintenance manual. In the figure, one blue box signifies one module. For clarity, in cases where a paragraph has to be divided, I have used multiple colors to illustrate the borders. The modules have been numbered for referencing.



1. Introduction

1 This manual describes how to operate and maintain the HI-FOG® Gas-driven Pump Unit (GPU). This manual aims to describe the GPU at a general level; note that the illustrations or details in this document may not reflect exactly the GPU model or version supplied if there are customer-specific modifications.

2 Any changes made to the supplied and installed system or its equipment are allowed only with the written consent from Marioff Corporation Oy.

1.1 Design

3 The Gas-driven Pump Unit (GPU) has been designed for the fire protection of enclosed spaces starting from small spaces of light and ordinary hazard occupancies up to large spaces of total compartment protection of special hazard spaces. The GPU protects enclosures up to 500 m³.

Typical applications are:

- Machinery spaces
- Combustion turbines
- Industrial equipment

4 The GPU uses high-pressure gas cylinders (air or nitrogen) and non-pressurized water cylinders (or other water supply). Pressurized gas (nitrogen or air) is used as the propelling and atomizing medium. The GPU system is activated by releasing the pressurized gas to start the pump module. The GPU releases through open HI-FOG® nozzles when activated. 5 It is self-contained and does not need any extra water or electric power to operate. 6 An electric signal is needed for automatic or remote activation and pressure or flow monitoring (optional).

7 The gas cylinders are equipped with Marioff gas valves. Typically one actuator gas valve is used to release the first cylinder and the slave valves in the other cylinders are released by the pressure released from the first cylinder. If multiple shots are used, the gas cylinder racks are released in predefined sequences by means of Delta-P valves.

1.2 Functionality

8 A signal from the fire detection system, a remote electric signal or pneumatic actuation, or a pull at a manual handle (optional) opens a valve to release gas which starts the pump module.

9 A pressure drop in the system is generated in two ways. A fire will cause one or several heat sensitive sprinkler bulbs break, opening the access of water through the activated sprinkler(s). Alternatively, the section valves are opened and water has access to the section through the spray heads.

Figure 2 Example of topical divisions in the GPU Operating and Maintenance manual

Modules 4 and 5 in Figure 2 illustrate the issue of dividing content without considering the reuse possibilities. The paragraph as a whole describes the basic qualities of GPU and could perhaps be used as such. However, while the standard GPU at the moment does not require electrical power as is suggested by module 5, it is in fact getting an option of electrical standby pump, which means that all standard GPUs cannot run without electrical power. Thus, the paragraph as a whole is not true for all GPUs, and cannot be reused as such. Module 4, on the other hand, is true about all GPUs and can be used in all GPU documents to describe the basic functions of the pump unit. Module 6 was separated to be used with GPUs that have the monitoring option, and thus need electrical power. This example not only proves that when dividing content, reuse possibilities have to be taken into consideration from the beginning, but also that some knowledge of the products is required. Performing a manual migration to the new documentation system, as is done at Marioff, allows for a more precise division of content. Carefully analyzing the content, and making these decisions aids in the future content reuse strategy. If the migration was done automatically, with a script, these kinds of issues illustrated by the example might be ignored when editing migrated content. Dividing content is part of developing the new information architecture for the content, since it will define how content can be reused in the future. As was stated in Section 2.1, information architecture lays the foundation for all future content.

Although identifying duplicate or near duplicate content is not part of Step 1 in the reuse analysis model, I had to consider duplicate content when dividing the documents into modules. This is illustrated by modules 5 and 6 in Figure 2, and the following paragraph from the Gas driven pump unit GPU data sheet:

> (1) The unit operation does not require any electrical power. Electrical power is applied for controlling, monitoring and signaling of the system performance.

The informational content is exactly the same as in the operating and maintenance manual (modules 5 and 6 in Figure 2), but it is worded a bit differently. With reuse, if duplicate content in one of the documents is divided in a certain way, it obviously will be divided the same way in other documents that include this content. However, the content needs to be re-worded in a way that it fits into other documents. As was discussed in Chapter 3, content in modular documentation has to fit any context. When the information about electrical power is divided into several modules, the content needs to be rephrased. See for example the first phrase of example (1): “The unit operation does not require any electrical power”. More information is needed for this phrase, because only GPUs which have a pneumatic pump and which do not use monitoring do not require any electrical power.

In the operating and maintenance manual, which is a longer document, the explanation about electrical power in different versions of GPU can be more comprehensive, but in the data sheets, a decision needs to be made if this more comprehensive explanation is included in the current data sheet or if the information needs to be rearranged, and several data sheets are created to include information about each of the variants. This example illustrates that content needs to be edited when moving to the modular documentation system, because content needs to fit any context, but also that dividing content may lead to the need to rearrange documents, and create different versions of the same document for different purposes. There are in fact several ways in managing duplicate content, besides editing, and I will discuss them in the next part of the analysis (Section 5.2).

Another observation that can be made from Figure 2 is that in many cases, the paragraph divisions in the operating and maintenance manual also became the module divisions. I have not divided the content much to sentence level, because it seldom seemed necessary. Only in cases in which the sentence kept recurring in the material has it been separated into a single module. For instance, the following sentences are recurring within and across the material:

(2) For more detailed instructions, check project-specific documentation.

(3) Only use spare parts that are manufactured, supplied or recommended by Marioff.

The above examples are common phrases in other Marioff documents as well, and for this reason it is useful to make them into individual modules to be reusable on their own. The fact that many paragraphs in the operating and maintenance manual function as an individual module illustrates that the writing style applied at Marioff already supports writing modular documentation. Although nothing can be said about the whole library since only one manual was analyzed in this study, this proves that there can be much more content in non-modular documents that is usable as such in modular documentation than a company might expect. However, this does not mean that content should not be analyzed carefully before migration. For example, the paragraphs in the data sheets analyzed in this study are more likely to have multiple ideas combined. This is illustrated by the following example from the data sheet Gas driven Pump Unit GPU:

> (4) At each stroke, constant volumes of water and gas are discharged into the network. The operating pressure in the system is time-dependent: the pressure at the sprinklers or spray heads gradually decreases from (80 ± 15) bar down to zero depending on the number of open spray heads and gas cylinders.

The first sentence of the example is not directly related to the rest of the paragraph. The first sentence describes the operation of the pump, and the second the operating pressure in the system. In data sheets, this has probably been done to save space, and fit the content into as few pages as possible. The data sheet in question is only two pages and has hardly any white space around the images and text. In XML, this kind of combining is not beneficial from the point of view of reuse, because different topics need to be separate. For this reason, controlling document length cannot be very easily done in XML, which from the point of view of usability is a positive thing. For example, white space is used to lessen the cognitive load, and it is an important factor that helps the users to focus their attention and skim through the content, according to Alred et al. (2009, 299). In XML, line spacing, margins, and space around images are determined in the stylesheet. From the point of view of printing costs for the company, not being able to control document length can be a negative thing, but since documents are not usually read from beginning to end, extra pages should not affect usability in a negative way. The difference between the operating and maintenance manual and the data sheets proves that although some documents can include a lot of material that is ready to be moved to modular documentation environment, some documents might need a lot of work.

After considering the types of issues discussed above, the operating and maintenance manual was divided into a total of 117 modules, and the four data sheets were divided into 3–11 modules. I did not count images as individual modules, although they are always saved separately into the CMS, and are always potentially reusable. I counted images as part of the immediate context, for example steps. Table 5 presents the exact number of created modules. In addition to counting the total number of modules, I also marked and counted modules that are specifically related to GPU. I did this to discover how much of the information could potentially be used outside the GPU documentation, for example in other pump unit documents or any other already existing Marioff documentation.

Table 5 Number of modules in the documents

Document	**Total number of modules**	**GPU specific modules**	**Non-GPU specific modules**
Gas-driven Pump unit - Operating and Maintenance manual	117	44	73
Water tank positioning for GPU	3	3	0
Gas Driven Pump Unit GPU – GPU-6	11	9	2
Gas Driven Pump - Dual GPU	11	9	2
GPU with wall-mounted cylinders	11	8	3

From Table 5, we can see that the data sheets include much more GPU-specific information than the operating and maintenance manual. This can be explained by the fact that data sheets are shorter documents that include mostly technical data specific for the product they describe, for instance the dimensions of the pump unit (see an example of a datasheet in the Appendix). Operating and maintenance manuals, on the other hand, include content such as safety information that is not product specific, and is often the same across many documents. Below is an example of a generic notice:

(5) **Notice!** HI-FOG® system uses high-pressure water, which can be dangerous if the system is not properly installed.

This notice is included in all Marioff documents that include any instructions to install, maintain or operate the system or parts of the system. It is not in any way specific to GPU. In fact, most of the notices, warnings and cautions in the material are reusable in other Marioff documents. Out of thirteen notices, warning and cautions, only one was GPU specific:

(6) **Notice!** If the system includes gas or water cylinder accumulators, they will be released if the GPU is released. …

Other Marioff products do not generally use gas, hence, this notice is GPU-specific. However, if a module includes only GPU-specific information, it does not mean that it is not reusable at all. In addition to being used in GPU documentation, it can be included in system level documentation that includes different kinds of pump units, as well as project-specific documentation. However, across the whole library these modules are less reusable, because they describe only one product. The amount of general content in the operating and maintenance manual is much greater than GPU-specific content. A little less than two thirds is generic content, and the rest is GPU-specific content. From the point of view of reusability, this means that majority of the content is reusable in other contexts in addition to the GPU documentation analyzed in this study. For Marioff, this is a positive outcome because one of their wishes for the new documentation system was to maximize reuse.

5.2 Identifying duplicate content and editing (step 2)

The second step of the analysis includes identifying content that is identical or nearly identical. Table 6 shows the number of modules that are duplicate or near duplicate in each document. I decided to count reusable content in each of the documents individually, because I thought it would give a better idea of the reuse possibilities. In the operating and maintenance manual, the duplicate content is both within the document and across different documents. The data sheets, as they are a shorter document type, have only duplicate content across documents, not within.

Table 6 Duplicate or near duplicate content

Document	Duplicate or near duplicate modules	Total number of modules
Gas-driven Pump unit - Operating and Maintenance manual	33	117
Water tank positioning for GPU	0	3
Gas Driven Pump Unit GPU – GPU-6	10	11
Gas Driven Pump - Dual GPU	9	11
GPU with wall-mounted cylinders	2	11

The amount of duplicate content is much larger than I initially expected. A little less than one third of the operating and maintenance manual is duplicate content, and as was discovered earlier, about two thirds is generic, non-GPU specific content that can potentially be reused in other documents. The data sheets, for the most part, only include duplicate content with the maintenance manual or other data sheets. These findings suggest that my hypothesis of very restricted reuse possibilities is in fact false. However, this does not prove that reusability is so high across other Marioff documents. For instance, the data sheets analyzed in this study might include more descriptive text than other Marioff data sheets, which can result in there appearing to be more reusable, duplicate content between the operating manual and the data sheets than there actually was if the whole documentation library was analyzed.

When all the duplicate content is identified, there are several options to manage it when moving to the modular documentation system (Bellamy et al. 2012, 184). Here are three of the most relevant ones for this study (ibid.):

1) Edit content to make it identical
2) Use conditional processing
3) Use content references (conref)

The first option is applicable when the information is exactly the same but written a little differently. Below is an example from two of the data sheets to illustrate such a case:

(7) The pump starts when the pressurized gas flows to the pump. A standby pressure of about 25 bar (362 psi) is maintained in the system by the pneumatic pump.

(8) The pump starts when the propelling gas flows to the pump. A stand-by pressure of about 25 bar is maintained in the system by a pneumatic pump.

The underlined parts highlight the minor differences between the content. As was discussed in section 3.2, one of the benefits of modular documentation is that content stays coherent. Combining and editing content, such as in examples (7) and (8), helps to maintain the coherence. This is beneficial for reuse purposes, but it also helps the user recognize certain paragraphs from separate documents, and form patterns of the informational content. According to Bellamy et al. (2012, 230), inconsistency can be distracting for the user and thus decrease usability. Especially Marioff employees, who regularly use the same document types, can begin to recognize what information paragraphs include by their wording and terminology. As was discussed in Section 3.2.2, making terminology more consistent also benefits the localization.

The second option, namely using conditional processing, works when there are different versions of the same product and the versions have a certain amount of mutual content and some differing content. In these cases, the same map can be used to publish documents for both of these versions. For example, content for the standard GPU and the Dual GPU could be included in the same map. The differing content would be marked, for example as “Dual GPU”, and when publishing with the right filter, the document would include the appropriate information for Dual GPU only. However, at Marioff, the CMS restricts conditional processing. The publishing is done within the CMS, and all filters have to be predetermined in the publishing engine. When a document is published, the publishing format is chosen from a dropdown menu. This means that for each filter there would have to be a separate format in the dropdown menu. Extensive use of conditional processing would quickly increase the number of items on the format list so that it would no longer be usable for the writers. In addition to this, the filters cannot be created by the writers, but an update request needs to be sent to the software provider. However, this might be a good thing for writers who are just beginning to use a modular documentation system, as is the case at Marioff, since according to Bellamy et al. (2012, 178), extensive conditional processing and filtering can over-complicate content maintenance.

In addition to keeping it simple, if the filtering is restricted, content can be assessed carefully before applying any conditional processing, a proper strategy for filtering can be made, and the filters that are actually needed can be listed for future updates to the system. On the other hand, as the CMS is part of the information architecture, this restriction with conditional processing proves that poorly developed information architecture indeed restricts information design and content creation, as was illustrated by the bathroom example in Chapter 2. When conditional processing cannot be easily done, the content will be reused in a different manner. Near duplicate topics and maps need to be created and content referencing to be used instead of conditional processing. As more and more content is migrated to the documentation system, the more laborious it will become to change the reuse strategy, because all the near duplicate modules have been taken into the system and are reused in different contexts.

The third option, namely using content references (conref), is a method that can be utilized to manage near duplicate content. This is a method that Bellamy et al. (2012, 178) suggest to be used as an alternative for conditional processing. For example, in the GPU operating and maintenance manual, there are five valve resetting instructions that have almost identical steps but some valves have some additional or slightly differing steps. So instead of creating filters for each valve, all the steps and images in the instructions can be gathered into one topic module. Figure 3 is a simplified illustration of how content referencing could be used with step instructions.

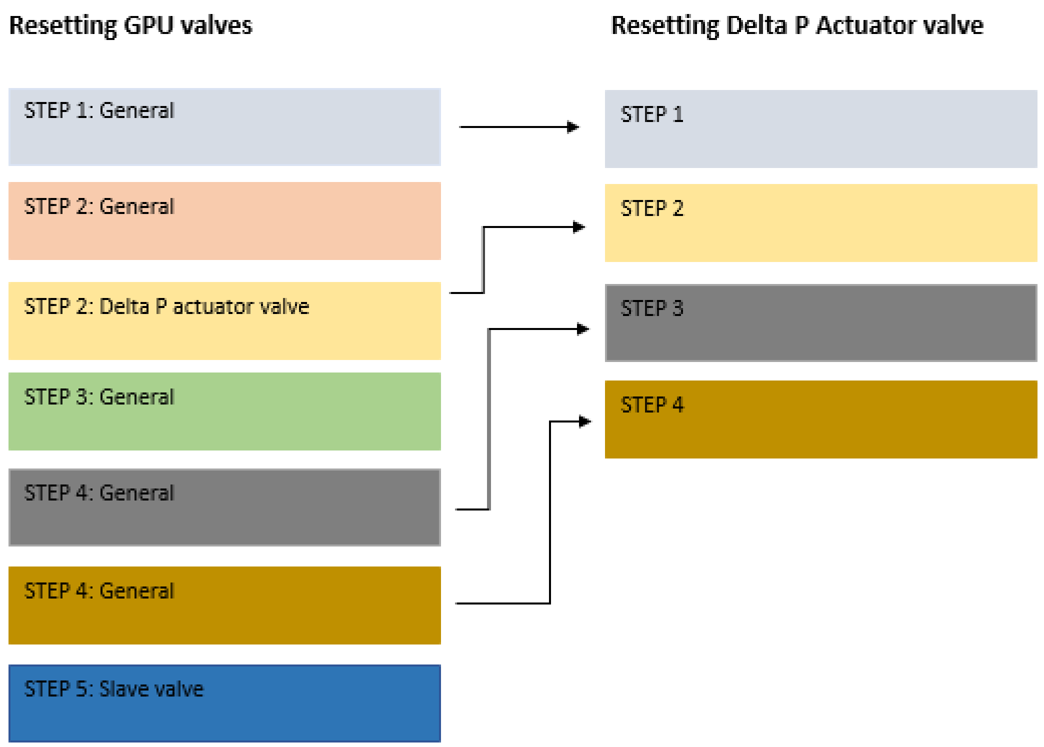


Figure 3 Using content referencing to manage duplicate content in step instructions

With the valve resetting example, *conref* works especially well, because all the steps can be gathered into one topic module in the right order with the differing steps marked (cf. conditional processing). This helps in managing the content, since if each step was done with an individual module, managing the right order of the steps would be cumbersome for the writers. From this common topic module, where the steps have been gathered, the steps can be reassembled into any order, or inserted in any other step list. As is seen from Figure 3, for example steps 2 and 3 are not included in the task of resetting Delta P actuator valve. As was discussed in Chapter 3, one of the benefits of modularity is the quick reorganization of information. Content referencing is one way to quickly reorganize content. For content referencing to work, there has to be instructions for the writers on how to create unique IDs for elements that are then easy to find and insert into different documents. As was stated by Ament (2007, x), standards are needed in modular documentation.

In addition to editing duplicate content to become more coherent, in some cases content has to be edited to become more general, in order for it to be reusable in different contexts. This is illustrated in the following example:

(9) This manual describes how to operate and maintain the HI-FOG® Gas-driven Pump Unit (GPU). This manual aims to describe the GPU at a general level; note that the illustrations or details in this document may not reflect exactly the GPU model or version supplied if there are customer-specific modifications.

If the underlined items are replaced by the more general term *pump unit,* this module can be reused in all pump unit operating and maintenance manuals. This is one of the writing guidelines for modular writing: all references to product names should be avoided if possible to keep modules reusable, according to Bellamy et al. (2012, 196). However, some content is product specific and does not need to be stripped of product names. For example, GPU is the only pump unit at Marioff that uses air or nitrogen as an atomizing medium, thus modules that describe these features related to gas do not need to be made more general by removing references to GPU.

In some cases, it can even be harmful for the informational value of the content to make it more general. Bellamy et al. (2012, 184) say that there is the danger of getting too eager to make everything reusable everywhere, eliminating all duplicate content. In the operating and maintenance manual, the valve resetting instructions are a good example of content that should not be made too general. As was discussed above, the resetting instructions for the different valves include many identical steps, but there are also steps that are slightly different. Examples (10) and (11) below illustrate this issue.

(10) Test the valve operation by pulling the manual release handle. The manual release sleeve moves smoothly up and down.

(11) Test the valve operation by pulling the manual release sleeve with a manual release tool (Marioff stock code D24215). The manual release sleeve moves smoothly up and down.

The underlined parts illustrate the difference between these steps. Although the steps are almost identical, the differences are very important and need to be taken into consideration when planning their reuse strategy. If the steps above are combined, and both testing methods, namely handle and release tool, are mentioned, it may result in the user being confused about how to complete the step. Mentioning a tool where it is not needed disrupts the information processing, and slows the action, because the user will stop to search for the tool or is confused whether they need the tool or not. Making the steps more general by not mentioning the handle or the tool is equally confusing, especially if the tool is needed for completing the step, because the user might not know how to test the valve operation. In these kind of cases, near duplicate content needs to be maintained.

One additional task to complete when moving to the modular documentation environment is to rewrite content to make it more uniform. As was discussed in Section 3.2.2, one of the main benefits of modularization is coherence, but to achieve coherence, there have to be writing guidelines. While analyzing the material, I also did some limited textual analysis, and marked discrepancies in terminology. A good example of the discrepancies in terminology was how titles and the product names were written. This is visible from the three document titles below:

(12) <u>Gas-driven Pump unit</u> – Operating and Maintenance Manual

(13) <u>Gas Driven Pump</u> – Dual GPU

(14) <u>Gas driven Pump Unit</u> GPU – GPU-6

There are three differences in the product names. The first one is hyphenation of the compound *gas driven*. The second difference is the capitalization of words in the titles. The third difference is the full name of the product: Gas driven pump unit, or gas driven pump. Earlier it was discussed that the SFS-EN 82079-1 (2012) defines that terminology should be coherent across product documentation to increase understanding (Section 3.2.2). The first two of the differences, namely hyphenation and capitalization, are not threatening understanding, but the third one is. Gas driven pump and gas driven pump unit could be understood as two different things, although they are used here to refer to the same product. This kind of discrepancy may confuse users. However, discrepancy in terminology is not only confusing for users, but can also give a bad image of the company, and damage their brand.

Hyphenation and capitalization are good examples of this. The whole example illustrates that an up to date style guide is needed to maintain coherence on the textual level.

Another thing that has to be edited when moving to modular documentation is locational referencing. In linear writing, it is typical to reference to something that was mentioned earlier or will be discussed later in the text. In modular writing, modules are not fixed to the context of what is before and what is after in the document. Any kind of relational language that points to another location in the document has to be edited, for example:

(15) See the figure below.

If the context changes, “a figure below” might be above in the next output of the document (Bellamy et al. 2012, 196). These kinds of locational references were quite rare in the material, but other kinds of references were found that need editing. For example, data sheet GPU with wall mounted cylinders includes the figure below (Figure 4) that has different parts of the system numbered. These numbers are referred to from other parts of the document (see Figure 5).

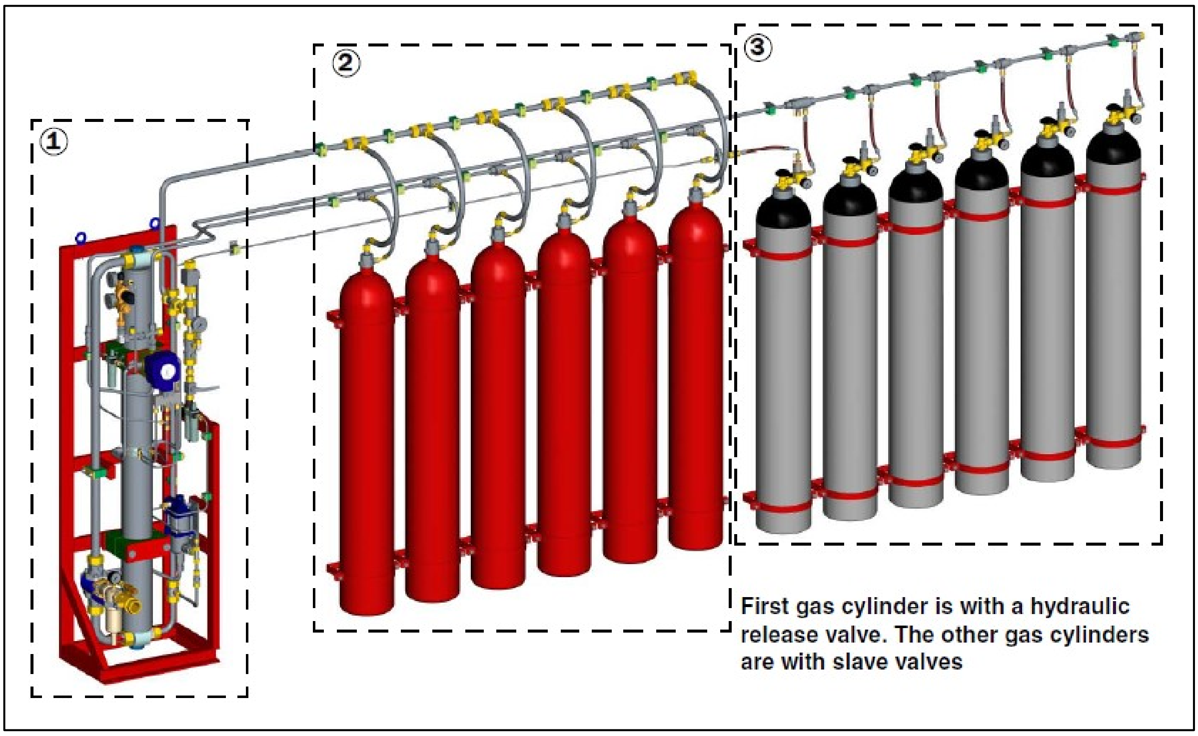


Figure 4 Referencing example from GPU with wall mounted cylinders

Pump module (item ① in Figure 1)

Stock code	Model	Monitoring	Additional information
E61045	GPU6D/A	No	Marioff Technical Data Sheet TE6076
E61045.4	GPU6D/B	Yes	Marioff Technical Data Sheet TE6076
E61046	GPUW/A	No	Marioff Technical Data Sheet TE6076
E61046.1	GPUW/B	Yes	Marioff Technical Data Sheet TE6076

Figure 5 Image referencing example

In modular documentation, this kind of referencing is not possible, because it restricts the reuse possibilities of both the image and the content that refers to the image. Although it is very illustrative and usable to have numbering across a document, without the numbering, the image and the table could be reused separately in multiple contexts. This content can be made more reusable by removing the numbering, and instead clearly marking the different parts of the pump unit in the figure title, and referring to the parts of the pump unit by their name (pump unit, water unit, gas unit). This kind of editing brings up the question of usability. In particular, does modularity reduce usability, for example when the referencing to parts in images cannot be done in the same way as illustrated above? Not necessarily, since not being able to use one way can lead to coming up with a better way. One option in this particular case is to add an image of each of the parts next to the relevant table. Although this makes the document longer, it allows the user to see the part in the immediate context of the table, instead of going back and forth between pages. This time the information architecture that is embedded in modularity affects the information design process. Content needs to be modified and developed so that it works in the modular documentation system.

In technical documentation, it is common to use tables to present complex information (Alred, et al. 2009, 299). Tables are easier to skim through than body text. For this reason, I also marked some content in the material that could be presented in table format instead of body text. These included any technical data inside body text, for example standby pressures in the data sheets and tool sizes in the operating and maintenance manual. This kind of analysis is one step in deciding the topic types for content. Information in table form is usually reference information as was illustrated in Section 3.1.1. Hence, when moving to the new documentation system, content should not just be copy-pasted, but information design should be created to fit the function of the information while also taking into consideration the topic types commonly used in DITA XML.

5.3 Guidelines for analyzing and restructuring content

After the content has been divided into reusable topics, duplicate content and content that needs editing to follow the conventions of modular documentation has been identified, and a strategy for managing duplicate content has been mapped out, some suggestions for analysis arise from the results. I will introduce the suggestion for analyzing material, and argument why each step is relevant in restructuring content. Strict guidelines for restructuring and reusing information cannot be made based on this analysis, which proves Bellamy et al.'s (2012, 196) similar statement that there is no magic tool to transform content into reusable topics. The following suggestions for analyzing material, illustrated in Table 7, are quite general, but they give an idea of how the analysis process could develop, and what needs to be taken into consideration when restructuring content.

Table 7 Guidelines for analyzing and restructuring content

Step 0. Prerequisites • Learn about the modular documentation solution the company is acquiring.
Step 1. Go through the content (keep in mind the whole library) • What other document types are currently used? • Who uses the information? • What outputs could be designed in the future?

Step 2. Divide a document into topics • Keep in mind product versions, other similar products, etc. • Consider preliminary reuse possibilities.
Step 3. Identify generic and product specific information **A.** For generic content, list all contexts where the information could be reused, and if needed, edit the content to become even more generic by removing product names. **B.** For product specific content, consider if the content is only for that document or can it be reused in other contexts for that particular product.

Step 4. Identify duplicate and near duplicate content
• Create a reuse strategy for the duplicate content that fits your system configuration, for example:
    o Editing and combining
    o Conditional processing
    o Content referencing

Step 5. Mark content that needs editing • Textual edits (terminology, complex sentences, etc.) • Referencing (locational, image references, etc.) • Information types (tables, lists, etc.)
Step 6. Go through the divided topics multiple times and re-divide if necessary

An important step before beginning to analyze content is to learn about the new documentation system solution that the company has obtained. Based on this study, the solution can affect the analysis, content creation, and reuse strategy. In Marioff's case, this means mainly the restrictions by the CMS to conditional processing and metadata usage.

The first actual step of the analysis is to familiarize oneself with the content. The step is also found in Bellamy et al.'s (2012, 197) reuse analysis model, and it functions as the base of the analysis. The order of the analysis can be adjusted to the needs of the project, but Step 1 should always appear early in the analysis. The purpose of the step is to better understand the larger picture, and the relationships between information before focusing on individual documents or topics. It will help to make decisions about dividing content later in the analysis. Considering current document types and thinking about any future documentation developments will provide a framework of where the documentation is now and how it should be modified and created in the future to reach future endeavors. Another matter that should be considered already in the beginning is the user groups for the content. As has been said many times during this study, content should always be created while considering the user. Document types, usage and users should all be kept in mind throughout the rest of the analysis.

The second step is to choose a document, or several related documents, and divide them into topics. This step requires some knowledge of the products, and other documents. For instance, in this study, knowledge about other Marioff pump units and their documentation helped make many of the topic divisions. If the first step is done carefully, the knowledge can be utilized in this second step. Reuse possibilities are most likely already appearing in this stage, so reuse should be kept in mind when dividing content. For example, when considering whether to divide something into two separate topics, it is worthwhile to think about their reuse possibilities: Is there a scenario where one of these pieces could be used in another context, but only if the other piece is not included? Coming up with concrete examples can be helpful when deciding the divisions.

The third step of the analysis is identifying generic and product specific information. Generic information can be hidden, and can seem like product specific content with mentions and references to one product. However, other similar products should be considered, and if the content could be modified and used with those other products. If content is strictly product specific, similar to topics about handling gas in the GPU documents, it is not necessary to make them more generic. During this step, it might be helpful to identify concrete examples where the content should be reused, and list them for future referencing.

Step four is to identify duplicate or near duplicate content. This step is also found in Bellamy et al.'s (2012, 197) reuse analysis model. Duplicate content might have already come up in other parts of the analysis, but it requires further attention. There are different types of duplicate content, and their reuse strategies vary accordingly. For example, for content that has identical informational content, but that is worded differently, simple editing is enough to make it reusable. For near duplicate content either conditional processing or content referencing can be used, based on the amount of mutual content in a document, and the context to which the pieces are applicable. For example, if at least half of the content is mutual, and the differing content is related to different products, conditional processing is more effective. If, on the other hand, most content is different, the mutual content can be inserted into the document by content referencing.

The fifth step of the analysis is marking content for any kind of editing. This can include textual edits, such as fixing discrepancies in terminology; adjusting referencing to be more suitable to the modular documentation environment, for example removing locational references; or making any information design decisions regarding the form or function of information, such as using tables to present technical data. This is an important step to ensure that the content being moved to the system is indeed written according to the standards of modular documentation, and is thus reusable.

The last step of the analysis is going through the material again, and checking the decisions made about the divisions and reuse strategies. This step can also include testing the new documentation system and verifying that everything works as was planned from the point of view of content creation.

6 Conclusions

In this study, I have examined how to best analyze and restructure technical documentation when moving to an XML-based documentation system. I did this by analyzing technical documentation from Marioff Corporation Oy, providing a restructuring of their documents, suggesting how their content could be modified to become more reusable, and creating a reuse strategy for reusable content. My conclusion based on this analysis was that universal guidelines on how to restructure information cannot be created, but some suggestions on how to analyze content can be provided. Bellamy et al. (ibid., 196) actually state that there is no magic tool to migrate content into a modular documentation environment, because content is always different. Based on this study, this statement can be said to be true. Especially with manual migration, the division of content relies much on the content, as well as the person analyzing the material. This is especially true when first starting the migration. This study was the first proper analysis of the Marioff documentation before migrating to the new system, which means that there were no concrete guidelines to follow. Knowing the product families and individual products was very important when considering the module divisions.

Based on the analysis, I created a guideline on how to proceed in restructuring information. The guideline is not fully comprehensive, because it is based on one analysis. However, the guideline can help avoid issues with reusability, and accommodate to the possible restrictions of a predesigned information architecture. The guideline is more suitable when preparing for manual migration of content. Since the analysis requires knowledge of the products and that the material is gone through very carefully in the analysis, the guideline is perhaps more suitable for analyzing a modest number of documents.

The hypothesis of this study was that there is not much to be reused in Marioff technical documents. However, this was proven to be untrue, and the amount of reusable content both across the material, and potential reuse in other documents was much greater than expected initially. This is a positive outcome for Marioff, and illustrates that analyzing content carefully can show the full reuse potential of content in any organization. However, the amount of reusable content in the analyzed material is not a guarantee that all Marioff documents will have as much to reuse. The material chosen, especially the data sheets, might have distorted the results by having more descriptive text than other Marioff data sheets. For this reason, conclusions about the full reuse potential cannot be made based on the results.

Theory-guided content analysis was a suitable method for this study, in my opinion, because the focus was on the material, but the theoretical background helped the argumentation and decision making of how the content is divided and why. The theoretical framework also gave a perspective on editing content to become more usable. However, the reuse analysis model by Bellamy et al. (2012, 197) as such was quite general even to be used as a guideline. It does give an indication of what steps need to be taken, but more comprehensive instructions on how to complete the steps would benefit the model – even if these instructions are only examples.

As there are no strict guidelines on how to divide content into topics, this study is just one example on how this particular material could be divided. The divisions are based on suggestions on how to achieve maximum reuse and how to edit content to make content reusable in any context. The actual divisions are my interpretations of the guidelines and suggestions, as well as what I have learned while working at Marioff, and what the company's hopes are for the new system. Since the documentation system did not come fully into use during this study, I was not able to test the whole new information structure for the analyzed documents.

The restrictions that were caused by the content management system proved how information architecture can affect the information design process. For example, restricted availability of filters forces the use of content referencing instead of conditional processing at Marioff. This will affect designing new content and its reuse in the future. Hence the conclusion is that information architecture and information design should indeed be developed together. Of course, in companies this is not always possible since other content, except technical documentation, might be managed in the same system, and there is no other way than to adjust to the existing infrastructure, and create content that works in the current system. Actually, modularity in itself also sets some limitations to information design since content needs to be created in a certain way to be reusable. An example of this is referencing inside a document. However, this is a limitation that is accepted when deciding to move to the modular documentation environment.

Modularity affects the whole authoring process, but also the review process. An interesting further study idea would be to study how the review process of modular documentation, in other words sending smaller pieces for review, changes the nature of feedback. In linear documentation, reviewers can comment on how chapters and paragraphs work together and the general structure and appearance of the document. In non-linear documentation, reviewers only look at individual topics and the accuracy of the content. Will usability of documents decrease when feedback about overall structure and appearance is not given in the review, but needs to be given separately after actual document use, or should other feedback gathering forms be developed in addition to the review process of technical information?

What was interesting to discover when analyzing the material was that technical documentation written for hardware products, and especially fire and security products, was already written in a manner that allowed content to be easily divided into topics. This flexibility promises further documentation development in the future, such as combining instructions with software, or making animations with already existing instructions. With keeping the future endeavors in mind, migrating into modular documentation system should be done with careful planning and patient analysis of the documentation library.

7 Works Cited

Albers, Michael J. 2003. Introduction. In *Content and Complexity. Information Design in Technical Communication*, Albers & Mazur, 1–13.

Albers, Michael J. & Mazur, Beth, eds. 2003. *Content and Complexity. Information Design in Technical Communication*. Mahwah, New Jersey: Lawrence Erlbaum Associates.

Alred, Gerald, Charles Brusaw & Walter Oliu. 2009. *Handbook of Technical Writing.* Boston: Bedford/St. Martin's.

Ament, Kurt. 2007. *Single-sourcing. Building modular documentation*. Norwich: William Andrew Publishing.

Ames, Andrea & Corbin, Michelle. 2007 "Information Architecture: Contributing Strategically to the Success of Our Customers and our Businesses." *Technical Communication*, 54, 1: 11–15.

Bellamy, Laura, Carey, Michelle & Schlotfeldt, Jenifer. 2012. *DITA Best Practices – A roadmap for Writing, Editing, and Architecturing in DITA*. Indianapolis: International Business Machines Corporation. Pearson plc.

Bernhardt, Stephen A. 1993. "The Shape of Text to Come – The Texture of Print on Screens". *College Composition and Communication*, 44, 2: 151–175.

Broberg, Mats. 2016. "A Decade of XML – and a New Procurement and Lessons Learned." *Technical Communication,* 63, 1: 6.

Carliner, Saul. 2003. "Physical, Cognitive, and Affective: A Three-part Framework for Information Design." In *Content and Complexity. Information Design in Technical Communication*, Albers & Mazur, 39–58.

Crystal, Abe. 2007. "Facets Are Fundamental: Rethinking Information Architecture Frameworks." *Technical Communication*, 54, 1: 16 – 26.

Eble, Michelle F. 2003. "The effects of single sourcing on the teaching of technical communication." *Technical communication,* 50: 344–349.

Eskola, Jari. 2015. "Laadullisen tutkimuksen juhannustaiat". Valli, Raine & Aaltola, Juhani. *Ikkunoita tutkimusmetodeihin 2*. Juva: Bookwell Oy. 185–206.

Evia, Carlos, Virginia Tech & Michael Priestley. 2016. "Structured Authoring without XML: Evaluating Lightweight DITA for Technical Documentation". *Technical Communication*, 63, 1: 23–37.

Eysenck, Michael W. & Keane, Mark T. 2000. *Cognitive Psychology: A Student's Handbook.* 4th ed. Cambridge: Taylor & Francis.

Ford, Julie Dyke & Mott, Richard K. 2007. "The Convergence of Technical Communication and Information Architecture: Creating Single-source Objects for Contemporary Media". *Technical Communication*, 54, 3: 333–342.

Ganier, Franck. 2004. “Factors Affecting the Processing of Procedural Instructions: Implications for Document Design.” *IEEE Transaction on Professional Communication*, 47, 1: 15–26.

Garrett, Jesse James. 2001. Response in “What’s in a name?” *STC Design Matters*, 5, 3. Online at: http://web.archive.org/web/20030210052847/http://www.stcsig.org/id/whatis.html#garrett. [Accessed January 18, 2018]

Graves, Heather & Graves, Roger. 2011. *A Strategic Guide to Technical Communication*. Alberta: Broadview Press.

Hackos, JoAnn. 1999. Documentation databases. Boston, MA: *Help Technology Conference. Solutions*.

Haynes, David. 2018. *Metadata for Information Management and Retrieval*. London: Facet Publishing.

Hoffmann, Maxwell. 2012. “A Review of Top 10 Trends Driving the Technical Communication Industry”. *Adobe Technical Communication Blog*. Online at: http://blogs.adobe.com/tech-comm/2012/04/a-review-of-top-10-trends-driving-the-technical-communication-industry.html [Accessed March 30, 2018]

Johnson, Tom. 2012. “Unconscious Meaning Suggested from the Structure and Shape of Help”. *I’d rather be writing*. Online at: http://idratherbewriting.com/2012/07/18/unconscious-meaning-suggested-from-the-structure-and-shape-of-help/ [Accessed February 26, 2018]

Kimbal, Miles A. 2017. “The Golden Age of Technical Communication.” *Journal of Technical Writing and Communication*, 47, 3: 330–358.

Knemeyer, Dirk. 2004. “Richard Saul Wurman: The InfoDesign interview.” Online at: http://www.informationdesign.org/special_wurman/ [Accessed 27 December 2017]

Koikkalainen, Tanja. 2002. *Single sourcing: a system for reusing information in documentation*. Master’s Thesis. University of Tampere.

Linton, Jenifer & Brunski, Kylene. 2006. *Introduction to DITA*: *A User Guide to the Darwin Information Typing Architecture*. Colorado: Comtech Services, Inc.

Mazur, Beth. 2003. “Information Design in Motion.” In *Content and Complexity. Information Design in Technical Communication*, Albers & Mazur, 15–38.

Polit, Denise & Hungler, Bernadette. 1997. *Essentials of Nursing Research: Methods, Appraisal, and Utilization*. Michigan: Lippincott-Raven.

Price, Jonathan & Price, Lisa. 2002. *Hot text: Web writing that works.* Berkeley: New Riders Press.

Redish, Janice. 2000. “What Is Information Design.” *Technical Communication*, 47, 2: 163–166.

Rosenfeld, Louis & Morville, Peter. 2002. Information architecture for the World Wide Web. 2nd ed. Sebastopol, CA: O’Reilly.

Sapienza, Filipp. 2004. ”Usability, Structured Content, and Single Sourcing with XML.” *Technical Communication*, 51, 3: 399–408.

Schriver, Karen. 2003. Foreword. In *Content and Complexity. Information Design in Technical Communication*, Albers & Mazur, ix–xii.

SFS-EN 82079-1 2012. *Preparation of instructions for use – Structuring, content and presentation – Part 1: General principles and detailed requirements.* Helsinki: Finnish Standards Association SFS.

Tuomi, Jouni & Sarajärvi, Anneli. 2002. *Laadullinen tutkimus ja sisällönanalyysi*. Helsinki: Tammi.

Appendix

Example of a Marioff data sheet

HI-FOG®
water mist fire protection

Technical Data Sheet

GPU with wall mounted cylinders

TECHNICAL DATA SHEET K0012277 **REVISION** B **DATE OF ISSUE** 5 Aug 2016

This document describes the items you need to build a Gas driven Pump Unit (GPU). For more detailed information on the GPU installation, see Marioff Work Instruction K0012281.

See figure 1 below for an example installation of cylinders in a single row.

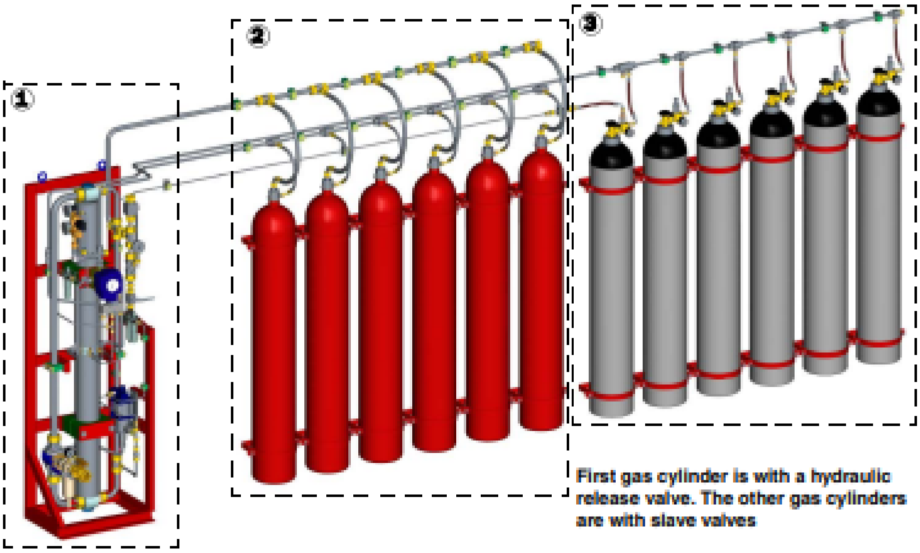


Figure 1: An example installation of cylinders in a single row. GPU items from left to right: (1) pump module, (2) water unit (red cylinders) and (3) gas unit (gray cylinders)

Marioff
United Technologies

GPU with wall mounted cylinders

Pump module (item ① in Figure 1)

Stock code	Model	Monitoring	Additional information
E61045	GPU6D/A	No	Marioff Technical Data Sheet TE6076
E61045.4	GPU6D/B	Yes	Marioff Technical Data Sheet TE6076
E61046	GPUW/A	No	Marioff Technical Data Sheet TE6076
E61046.1	GPUW/B	Yes	Marioff Technical Data Sheet TE6076

Water unit (item ② in Figure 1)

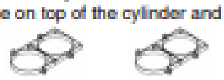
Stock code	Model	Additional information
E11157.6	WGU6/WM	Water cylinder assembly. See appendix figure K0010569.

Gas unit (item ③ in Figure 1)

Stock code	Model	Additional information
E62008	GPUCW/H	With hydraulic release valve. See appendix figure K0010386.
E62014	GPUCW/E	With electric release valve. See appendix figure K0010404.
E62010	GPUCW/DP	With Delta P release valve. See appendix figure K0010419.
E62012	GPUCW/SL	With slave valve. See appendix figure K0010390.

Items needed for installation

The following items must always be ordered separately. No piping or clamps is included in the pump module, water unit or gas unit. The minimum distance between two cylinders is 320mm which means the piping length needed between two cylinders is 500mm. The cylinders are always installed with two clamps on top and bottom of the cylinder. It is not allowed to install the water cylinders in more than two rows.

Stock code	Model	Additional information
A01005	DIN17457 D12x1,2W B	Welded piping to connect hydraulic valve
A01015	DIN17457 D16x1,5W B	Welded piping to connect water cylinders
A01025	DIN 17457 D25x2W B	Welded piping to connect water cylinders
E13010.2	D229 CLAMP SET WM	Clamp (single) to install cylinders in one row. Install the cylinders with two clamps: one on top of the cylinder and one on the bottom. 
E13011	D229 CLAMP SET	Clamp (pair) to install cylinders in two rows. Install the cylinders with two clamps: one on top of the cylinder and one on the bottom. 

Marioff Corporation Oy • Plaza Business Park Halo, Äyritie 24, 01510 Vantaa, Finland • Tel. +358 10 6880 000 • www.marioff.com

GPU with wall mounted cylinders

Compressed air supply for stand-by pump (items ① and ② in Figure 2)

Stock code	Model	Additional information
E10112		Stand-by cylinder assembly (item ① below), that is to be used only as a back-up solution for compressed air supply. See Marioff Technical Data Sheet TE6112.
M-0000347	400VAC, 50Hz	Stand-by pump assembly (item ② below). See Marioff Technical Data Sheet K0006197.
M-0000140	230VAC, 50Hz	
M-00005818	230VAC, 60Hz	

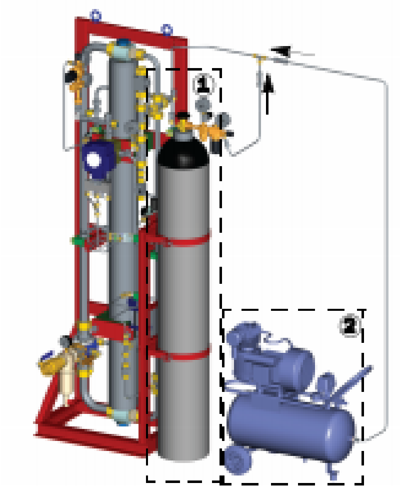


Figure 2: GPU with compressed air supply for stand-by pump as the primary solution with a stand-by cylinder assembly as a back-up solution

Marioff Corporation Oy • Plaza Business Park Halo, Äyritie 24, 01510 Vantaa, Finland • Tel. +358 10 6880 000 • www.marioff.com